MICR'ORIC

LE MAGAZINE DES UTILISATEURS D'ORIC

La palette
aux 4096 couleurs

ISSN-0759-9587 - Hiver 85-86 - N°10 30 F

BULLETIN D'ABONNEMENT

☐ *Je m'abonne aux 4 prochains numéros de MICR'ORIC .... **100 F***

(préciser ici à partir du n° ________)

*Pour compléter ma collection, je désire recevoir les numéros suivants : ________________ à **40 F pièce** port compris, soit : ____________*
(n° 1, n° 2 et n° 8 épuisés)

Ci-joint un chèque total de : ________________________________
*à l'ordre de **MICR'ORIC, Z.I. La Haie Griselle, B.P. 48, 94470 Boissy-Saint-Léger***

NOM : ______________________ Prénom : ______________

Adresse : ____

Ville : ______________________ Code postal : |_|_|_|_|_|

Date : ________________ Signature :
(des parents pour les mineurs)

EXTRAITS DES SOMMAIRES DES MICR'ORIC PRÉCÉDENTS

	ARTICLES TECHNIQUES ET CONSEILS PRATIQUES	Niveau de difficulté	JEUX ET PROGRAMMES	Niveau de difficulté
N° 3	— Comment tirer parti de votre ORIC-1	*	— Les carrés invisibles	*
	— Tracés en couleurs	*	— L'alcootest	*
	— Changement de numérotation	**	— Mille pattes, le serpent, ski	*
	— Hard copy d'écran hires	*	— Problème des 8 dames	**
	— ORIC et l'imprimante SEIKOSHA GP100	*		
N° 4	— La page ∅ Octet par Octet	***	— Carré magique	**
	— Un fréquencemètre	**	— Rubik cube	**
	— Verify pour ORIC-1	*	— Musique sous forth	**
	— Dessins hires assisté	*	— Snaforic	*
	— Pour stocker la page hires	*	— Course de chevaux	*
	— Diagrammes	*	— Mur de briques	***
			— Mission suicide	*
N° 5	— Les variables	***	— Dollar man	*
	— Un tampon pour imprimante	**	— Fort Oric	*
	— Disk search, initialisation	**	— Rase-mottes	*
	— Un merge pour ORIC-1	**	— Horloge	**
	— MCP 40 en mode graphique	*	— Des chiffres ou des lettres	*
N° 6	— Visite en tête de RAM	***	— Carwar	*
	— Dominez votre clavier	**	— Oric man	*
	— Bonjour les microdisques	**	— Solitaire en rectangle	*
	— ORIC en maternelle	*	— Carrés invisibles	*
	— Les adresses des fonctions	**	— Cocktails	*
N° 7	— Les variables (suite)	***	— La truite	*
	— Traceur de courbe	*	— Music	*
	— Renum	**	— Cross Oric	**
	— Hard copy MCP 40	**	— Cocktails (suite)	*
	— Maîtriser les couleurs	*	— Toutou	*
	— Détournement des interruptions	**		
N° 9	— Extension MULTICOLORIC (I)	***	— Romstoïca	*
	— Mini interpréteur MCP 40	*	— Julias	*
	— Un câble péritel à faire soi-même	**	— Cryptage	*

* DÉBUT'ORIC ** FORTICH'ORIC *** CRAC'ORIC



LE MAGAZINE DES UTILISATEURS D'ORIC

SOMMAIRE

N° 10

MICR'ORIC est une publication de la société A.S.N. Diffusion.

Directeur :
Jean TAIEB

Rédacteur en chef :
Lucien AUGUSTONI

Ont collaboré à ce numéro :
Dany BOSC
Jacques BRY
Stéphane CALLÉGARI
Dominique CHARMEAU
Pierre CHICOURRAT
Jean FINKA
Jean-Paul LEBAS
Pierre LEDAIN
Jacques-François WADEL

Adresse :
MICR'ORIC
Z.I. La Haie Griselle
B.P. 48
94470 Boissy-St-Léger

Dessins :
Alain TALVAT
Gilles TOCUT
Gérard BENKEL

1re couverture, dessins, créations et conception :
STUDIO MELUN-IMPRESSIONS

Imprimerie :
MELUN-IMPRESSIONS
18-19, rue E.-Briais, 77000 MELUN
Tél. : 64 52 04 31
ISSN 0759 9587

Editorial

Avec ce numéro 10 de MICR'ORIC les amateurs d'électronique vont pouvoir passer à l'action.

Quand ils auront goûté à cette palette de 4 096 couleurs, ils ne voudront plus s'en passer.

Si vous éprouvez des difficultés en cours de réalisation, l'auteur M. Jacques BRY s'efforcera de vous conseiller. Ceux qui, les premiers, sauront tirer un parti intéressant de cette interface nouvelle pourront faire paraître dans nos colonnes le reflet de leurs trouvailles.

Nous sommes heureux d'apporter à tous les utilisateurs d'ORIC un SUPER INPUT de 254 caractères acceptant les commandes usuelles du clavier. Personne, semble-t-il, n'avait songé à cette possibilité. Nous remercions M. Pierre LEDAIN de nous faire profiter de son travail : cela n'a pas été si simple ! Les chaînes saisies ainsi sont mémorisables sur disque ou sur cassette en couleur, double hauteur, etc... Nous connaissons une bibliothèque qui l'utilise, c'est éblouissant !

Stéphane CALLEGARI nous apprend à réfléchir à un problème : "comment remplir un contour fermé ?"

Il nous apporte une solution en BASIC et une autre en langage machine. Une suite au prochain numéro vous apportera un réticule pour viser la zone à colorier, cela explique le titre de cet article soigné.

Jacques-François WADEL vous apporte une méthode de pêche aux variables pleine d'attrait.

Un lecteur nous ayant soumis le problème du bulletin de notes, nous avons travaillé la question et nous proposons à tous ce programme. Pour ceux qui ne sont pas directement intéressés, nous en suggérons l'étude : l'abondant commentaire fourni vous donnera des méthodes de programmation.

Pierre CHICOURRAT, habile à manier le langage machine utilise des méthodes en BASIC qui sont analogues à celles propres au langage machine.

Par exemple le sous-programme lignes 390-460 donne une dynamique à la visite du fichier tout à fait comparable à ce qu'on voit sur des systèmes plus sophistiqués.

Jean-Paul LEBAS vous offre ses idées pour jongler avec les lecteurs de disquettes.

Dany BOSC a repris les idées de J.F. WADEL et apporte une version en langage machine, ceci devrait intéresser ceux qui veulent s'initier, le listing commenté nous semble très clair.

Nos deux jeunes dessinateurs continuent d'égayer la revue de leur généreux talent. Fin 86 sur Antenne 2, surveillez le générique des dessins animés, vous y verrez sûrement le nom de Gérard BENKEL, le voilà dessinateur professionnel sur ordinateur.

ORIC s'ouvre à la télématique avec des Modems divers et le LORITEL proposé par LORICIELS (utilisable avec un MINITEL), et ce n'est qu'un début.

ORIC, porté par l'enthousiasme de vous tous se porte très bien. L'année 1986 en apportera la confirmation. Grâce à vous, votre ordinateur favori ne cessera de nous combler. Excellente année.

MICR'ORIC



EXTENSION MULTICOLORIC (Suite)

Par Jacques BRY

(Vous trouverez le début de cet article dans MICR'ORIC N° 9).

III. - RÉALISATION

III.1. - Réalisation de la carte électronique

Le schéma électrique complet est donné **Figure 6 :** on y distingue deux circuits intégrés.

Les 5 premiers 74 LS 64Ø
25 LS 2521
74 C 221
EF 9369 (circuit palette)
TEA 5114 (ampli vidéo)

sont suffisants pour la réalisation d'une extension multicoloric 8 couleurs.

Les 6 autres 74 LS 02
74 LS 122
2 × 74 LS 163
74 LS 73
74 LS 373

permettent l'extension à 16 couleurs.

Le schéma comporte également les quelques composants passifs additionnels.

— Pour simplifier les câblages des alimentations "OV"(symbole ⏚) et "+ 5V"(symbole -+ 5V) n'ont pas été représentées.

— N.C. veut dire non connecté ;

— ◉ ce symbole représente les entrées sorties de la carte électronique vers l'*ORIC* ou vers le téléviseur.

La carte pourra se faire sur le support de votre choix : circuit imprimé ou carte à wrapper. Un format standard tel que 105 mm × 180 mm est convenable.

Carte, connecteurs et régulateurs d'alimentation tiennent dans un coffret standard 110 × 190 × 60.

— Précisons encore que l'on peut utiliser des circuits TTL-LS ou CMOS indifféremment.

III.1.1. - Décodeur d'adresse

Il est composé des deux C.I.

- 25 LS 2521 Comparateur 8 bits
- 74 LS 02 Quadruple NOR à 2 entrées

Le décodeur utilise la sortie I/O control disponible sur le connecteur d'extension d'*ORIC*.

Cette broche délivre un niveau "O" quand le microprocesseur adresse une position d'entrée sortie (page 3 de la mémoire *ORIC* soit # 3ØØ à # 3FF). Son utilisation permet de faire l'économie du décodage des adresses de poids fort# AD8 à AD15.

Il suffit maintenant de sélectionner des adresses dont l'usage ne perturbe pas le fonctionnement de l'*ORIC*. Nous avons choisi # 3F9 pour la mémoire du numéro de ligne de commutation entre les deux palettes, et # 3FF, # 3FD pour les deux adresses du circuit palette.

Le circuit 25 LS 221 est un comparateur logique 8 bits, sa sortie (broche 19) est active (niveau Ø), quand les entrées se correspondent bit à bit. Tel qu'il est câblé le 25 LS 221 décode les adresses.

3FF ⟶ N° de couleur
3FD ⟶ données couleur
3FB ⟶ non utilisé
3F9 ⟶ N° de ligne.

La séparation de ces adresses est effectuée par le 74 LS Ø2. Il est utilisé de la manière suivante :

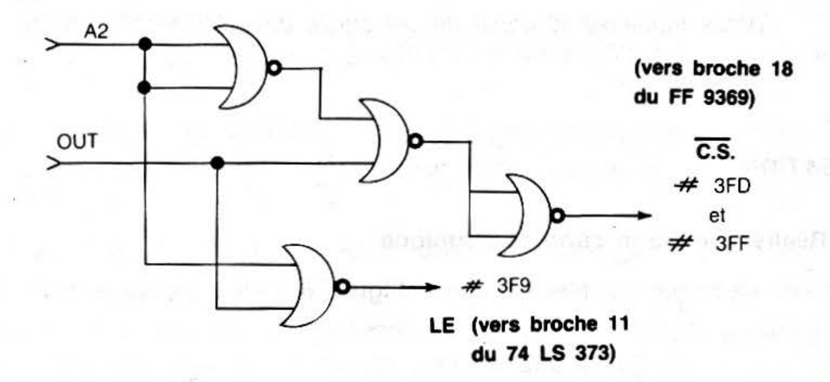


Figure 7

• **C.S.** est utilisé pour commander le circuit palette lequel dispose d'une entrée d'aiguillage (broche 22) à laquelle on connecte A1 ce qui permet la différenciation entre # 3FF et # 3FD.

• **LE** est utilisée pour commander l'écriture en mémoire du N° de ligne de séparation de l'écran.

Nota : Dans le cas d'une extension restreinte à 8 couleurs, le 74 LS Ø2 n'est plus nécessaire.

• Connecter A2 sur la broche 8 du 74 LS 2521 (en déconnectant la broche 8 de la masse).

• Déconnecter la broche 9 de la masse et la connecter au + 5V.

Le décodage est ainsi limité à # 3FD et # 3FF lesquels sont distingués dans le circuit palette EF 9369 par l'entrée A1.

III.1.2. - Buffer de données

C'est le 74 LS 64Ø qui remplit cette fonction. Il "isole" les données du bus Micro DØ à D7 de leur utilisation par les circuits internes à la carte. Normalement isolé le bus micro n'est ainsi connecté que lorsque l'une des adresses est détectée et durant la phase d'écriture (commande R/W) — (read/write), connectée à la broche 1 du buffer.

III.1.3. - Circuits de synchronisation

Rappelons nous la description du chapitre II.2. Les deux monostables M1 et M2 sont intégrés dans le circuit 74 C 221.

Deux condensateurs C1 = 6 nF et C2 = 1,2 nF permettent de générer les constantes de temps utiles en liaison avec les résistances ajustables R1 = R2 = 10 kΩ. C'est le réglage de ces deux résistances qui vous permettra de déterminer la fenêtre de blanking de votre choix sur le téléviseur.

Rappelez-vous qu'à l'intérieur de cette fenêtre vous pourrez programmer la couleur de fond.

Le signal SYM est l'un des deux signaux prélevés directement à l'intérieur de votre *ORIC*. Pour cela vous reporter à la description du chapitre III.2.

La sortie BLK du 74 C 221 (broche 12) est utilisée directement sur la broche 23 du EF 9369.

III.1.4. - Séparation synchro trame

Cette fonction optionnelle n'est utilisée que pour obtenir 16 couleurs ; elle est réalisée par le C.I. 74 LS 122. La sortie remise en forme (broche 6) est utilisée comme commande des circuits de comptage de ligne 74 LS 163 et 74 LS 73.

III.1.5. - Circuit palette

Comme nous l'avons vu c'est le cœur de votre extension. Il s'agit d'un circuit spécifique le EF 9369 réalisé en technologie de haute intégration. Composé de plusieurs milliers de transistors, il réalise l'ensemble des fonctions de décodage, de mémorisation et de conversion numérique - analogique. Nous n'entrerons pas plus dans le détail de son fonctionnement, ceci n'étant pas indispensable à son utilisation.

Décrivons ses différentes entrées-sorties.

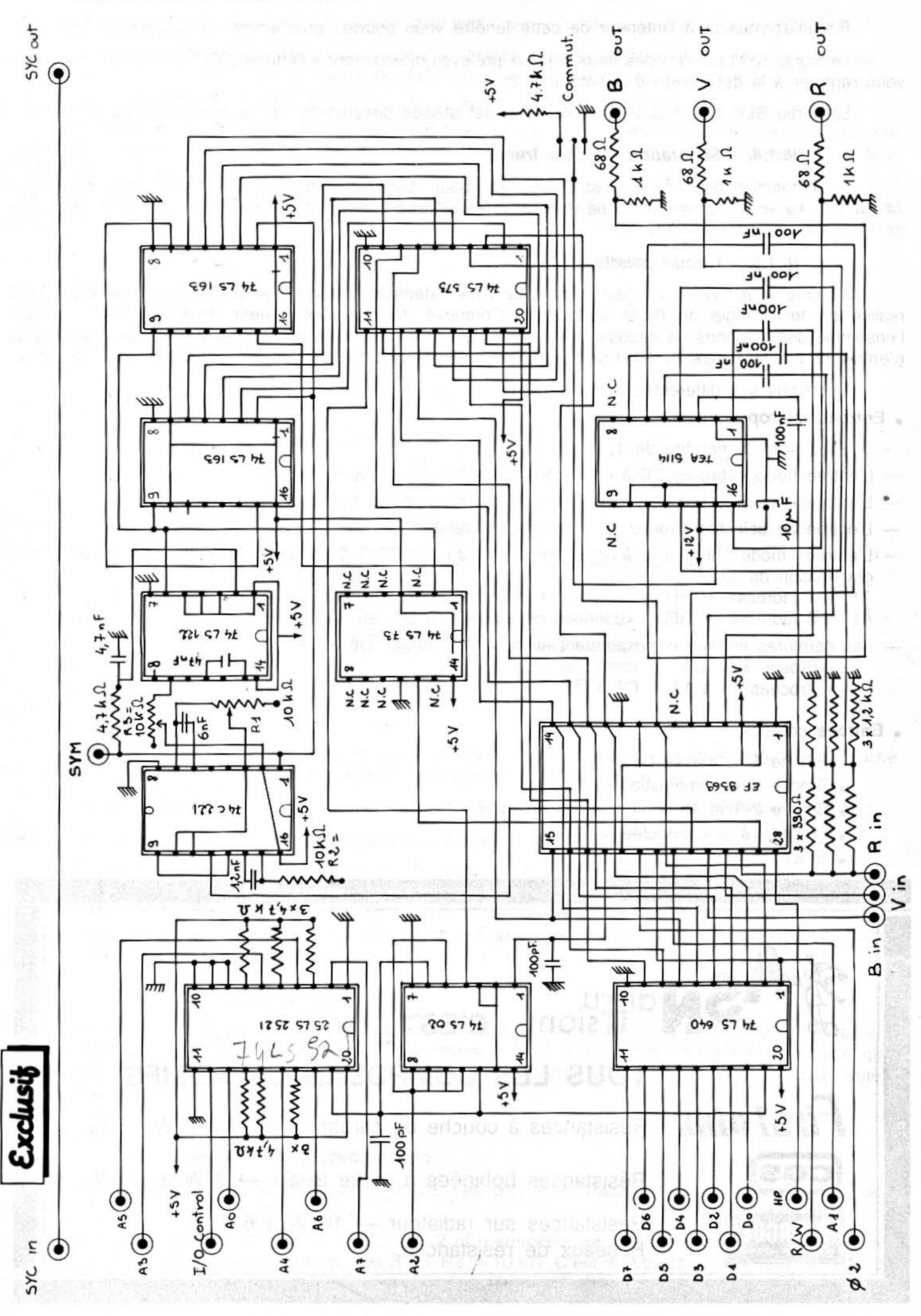
- **Entrées microprocesseur**

Elles sont au nombre de 12.

- L'entrée horloge broche 2Ø à connecter sur la broche Ø2 du bus *ORIC*.
- L'entrée read/write broche 21 à relier à la broche R/W du bus d'*ORIC*.
- L'entrée de sélection broche 18 à relier à la broche 1 du 74 LS Ø2.
- L'entrée "mode" broche 22 à connecter sur A1 du bus *ORIC*. Cette entrée sélectionne le cycle de programmation du circuit palette.
  A1 = 1 (adresse # 3FF) : choix du nombre de couleur (Ø à 15 sur DØ à D3).
  A1 = Ø (adresse # 3FD) : données couleur (DØ à D7) en deux cycles.
- Les dernières entrées correspondent au bus de données DØ à D7
  - Broche 8 = DØ.
  - Broches 11 à 17 = D1 à D7.

- **Entrées câblées**
  - Broche 1 = masse OV.
  - Broche 9 = alimentation + 5V.
  - Broche 2-3 et 19 = connectées à + 5V.
  - Broche 1Ø = connectée à OV.
  - Broche 7 = non connectée.

Figure 6. Carte électronique



• Entrées vidéo

— Broche 28 c'est l'horloge point, elle est issue de l'intérieur de l'*ORIC*, elle est utilisée sur le circuit palette pour synchroniser toute l'image.

— Entrées R, V, B (Rin, Vin, Bin). Broches 24, 25, 26. Elles sont directement raccordées aux sorties R, V, B de l'*ORIC*. (Veillez à bien respecter la correspondance sinon vous auriez des permutations de couleurs).

Notons que ces entrées sont également reliées au circuit ampli vidéo (TEA 5114) à travers un attenuateur (390 Ω -1,2 k Ω) pour permettre par commutation, la visualisation directe de l'image d'*ORIC*.

Entrée blanking broche 23 à relier au signal BRK broche 12 du 74 C 221.

— Entrée commutation palette broche 27 à connecter à la broche 12 du 74 LS 73, (à la masse dans le cas d'une extension limitée à 8 couleurs).

• Sorties vidéo

Ce sont les sorties des convertisseurs numérique - analogique qui déterminent trois nouveaux signaux R, V, B capables de prendre chacun 16 niveaux distincts. Leurs différentes combinaisons déterminent 4Ø96 teintes. Les broches concernées sont les broches 4, 5, 6 à connecteur respectivement aux broches 6, 4, 1 du TEA 5114 à travers un condensateur de 100 nF.

III.1.6. - Compteur de ligne et mémoire

Ils sont réalisés à l'aide des 4 C.I.

2 × 74 LS 163 - compteurs 4 bits.
1 × 74 LS 73 - bascule.
1 × 74 LS 373 - mémoire latch 8 bits.

Les deux 74 LS 163 sont des compteurs 4 bits binaires à préchargement que l'on cascade pour obtenir un compteur 8 bits.

A chaque trame ce compteur est préchargé à la valeur binaire mémorisée dans le catch 74 LS 373. Ce préchargement est commandé par le signal S.C. (voir Figure 3) de la broche 6 du 74 LS 122.

Le latch mémoire est programmé à l'aide de l'adresse # 3F9 décodée sur la sortie 4 du 74 LS Ø2 (signal **LE** Figure 7). Le contenu à mémoriser est la donnée DØ-D7 présente au moment de l'adressage en # 3F9.

Le compteur 8 bits préchargé s'incrémente à chaque ligne par l'intermédiaire du signal de synchro SYM. Quand il passe par zéro (débordement) il positionne la bascule 74 LS 73 broche 12 à "1". Cette bascule est remise à zéro par le préchargement.

Cette broche 12 est donc alternativement à "1" et à "Ø" pendant la durée d'une trame. La transition de "Ø" à "1" se fait sur le numéro de ligne programmé par # 3F9, elle effectue la commutation entre les deux palettes de 8 couleurs.

III.1.7. - Ampli vidéo

C'est le TEA 5114, il réalise l'adaptation complète des signaux vidéo entre le circuit palette et la prise péritel de votre téléviseur ou moniteur. Ce composant est alimenté sous 12 volts (performance oblige) et de ce fait il restitue parfaitement la netteté de l'image habituelle de votre *ORIC* ce qui ne serait pas le cas de n'importe quel "bricolage" vidéo à transistors.

Il dispose en outre d'un triple commutateur vidéo ce qui nous permet d'apporter gratuitement à l'extension un ultime perfectionnement. C'est la commutation entre l'image *ORIC* et l'image palette. Un simple interrupteur permettra de disposer des deux modes de fonctionnement sans avoir rien à brancher ou à débrancher. Cette commutation peut même se faire pendant le fonctionnement de l'appareil.

Les sorties **Rout, Vout, Bout** générées par le TEA 5114 remplacent les sorties R, V, B de l'*ORIC* il suffit d'y ajouter la synchro issue de l'*ORIC* en la transmettant par la carte électronique. Ceci est symbolisé sur le schéma Figure 6 par la liaison SYC in → SYC out.

III.2. - Intervention dans l'ORIC

Ne soyez pas trop inquiets, cette intervention est facile et sans danger.

Une fois l'*ORIC* ouvert il faut disposer une fiche **DIN** femelle 3 broches. Nous suggérons de la disposer sur le côté gauche du fond de boîtier (côté opposé à l'ouverture du bouton de reset). Celle-ci pour ne pas entrer en contact avec d'autres composants doit se trouver une fois en place, juste derrière le boitier métallique du modulateur U.H.F.

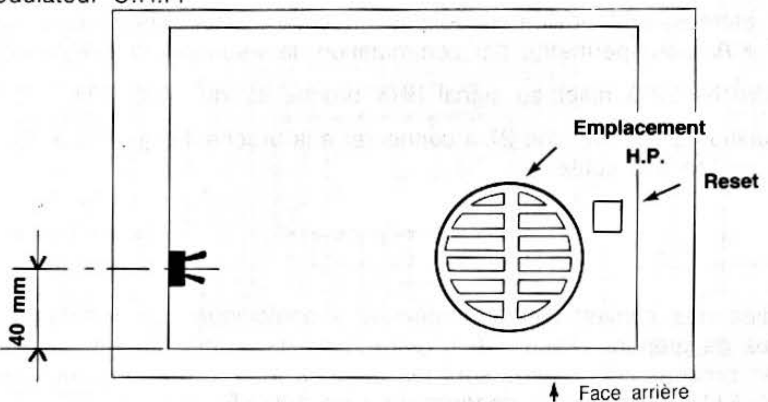


Figure 8

La photo n° 3 montre l'implantation de cette fiche qui se retrouve donc sur le côté droit de l'*ORIC*, une fois celui-ci refermé.

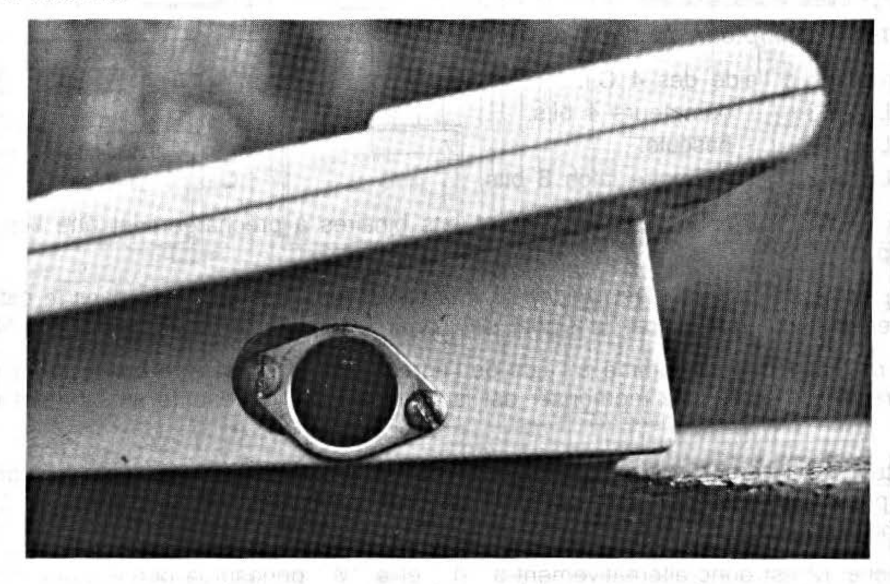
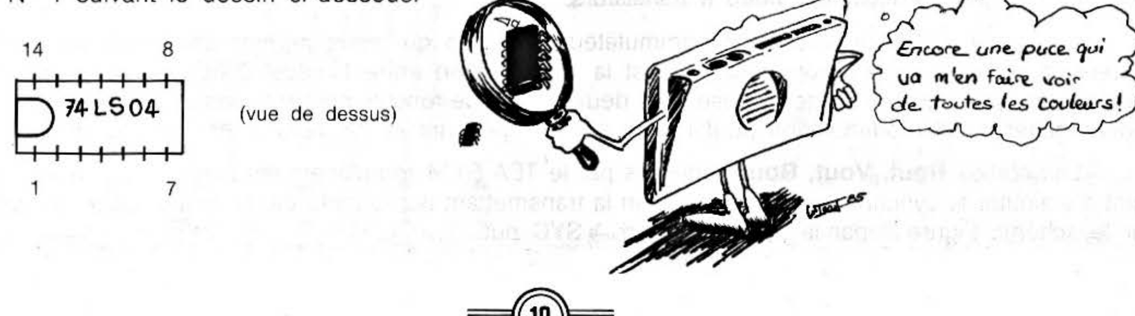


Photo n° 3

Par ailleurs un circuit intégré additionnel est nécessaire pour l'adaptation des signaux que nous allons prélever et notamment l'horloge point : H.P. C'est un 74 LS Ø4.

Regardez maintenant le circuit imprimé de l'*ORIC*, vous voyez en bas à gauche une rangée de 8 petits boîtiers identiques (ce sont les boîtiers mémoire **RAM**). Juste au dessus il y a trois autres boîtiers disposés horizontalement un 74 LS Ø4 et 2 × 74 LS 257.

Dans l'espace vide au dessus des deux premiers collez à l'araldite un petit rectangle de papier (il servira d'isolant) puis coller le 74 LS Ø4 sur le rectangle de papier, les "pattes" en l'air en repérant bien la broche N° 1 suivant le dessin ci-dessous.



La colle étant sèche établissez les liaisons suivantes (n'oubliez pas que le 74 LS Ø4 est à l'envers).
— Relier broche 7 du 74 LS Ø4 à la broche 7 du 1er 74 LS Ø4 (prise de masse).
— Relier la broche 14 du 74 LS Ø4 à la broche 14 du 1er 74 LS Ø4 (alimentation + 5V).

Voyez ensuite un grand circuit intégré de 40 broches juste à côté du haut-parleur. C'est le contrôleur graphique de l'*ORIC*.

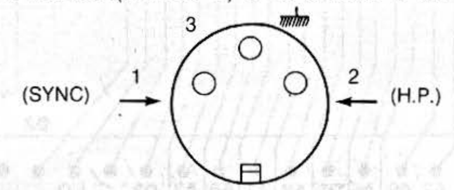
La broche 1 est repérée à gauche de l'encoche en vue de dessus.

Connectez sa broche 7 à la broche 1 du 74 LS Ø4 (c'est la fameuse horloge point).

Puis reliez la broche 16 du contrôleur graphique à la broche 1 de la fiche **DIN** précédemment installée (c'est la synchro).

— Reliez ensuite la broche 2 du 74 LS Ø4 à la broche 3 de la fiche **DIN.**

— Terminez en reliant la masse (broche 7) à la broche 3 de la fiche **DIN.**



Et voilà c'est fini vous pouvez refermer votre *ORIC*. Vérifiez bien que la fiche **DIN** en refermant le boîtier de l'*ORIC* trouve bien sa place et ne touche aucun élément du circuit (composant ou connexion).

III.3. - Raccordement de l'ORIC à la carte électronique

Le meilleur moyen d'effectuer le raccordement est de placer sur votre carte (circuit imprimé ou carte à wrapper) un connecteur mâle 34 broches coudé pour nappe de fil à sortir; (type TBA 6Ø934Ø7 ES). Ce connecteur ressemble à celui qui est disposé sur le bas d'extension de votre *ORIC*, mais il dispose de deux leviers d'extraction.

Il vous faut ensuite une nappe de fil à sertir, 34 brins, longueur 30 cm
— deux fiches **DIN** mâles, l'une de 3 broches, l'autre de 5 broches.
— deux connecteurs femelle 34 broches (type TBA 6Ø934ØØM).

Sertissez tout simplement votre nappe de 34 fils dans l'un des deux connecteurs femelle.

De l'autre côté il y a trois points de raccordement sur l'*ORIC*.
— *Le connecteur du bus d'extension.*
— *La fiche* **DIN** disposée en III.3.
— La fiche **DIN** 5 broches d'origine (sorties R, V, B).

Seuls 27 fils seront utilisés sur les 34.

Suivez maintenant le plan de raccordement **Figure 9** sur lequel nous préconisons les affectations de N° de broche du connecteur mâle de votre carte électronique d'extension.

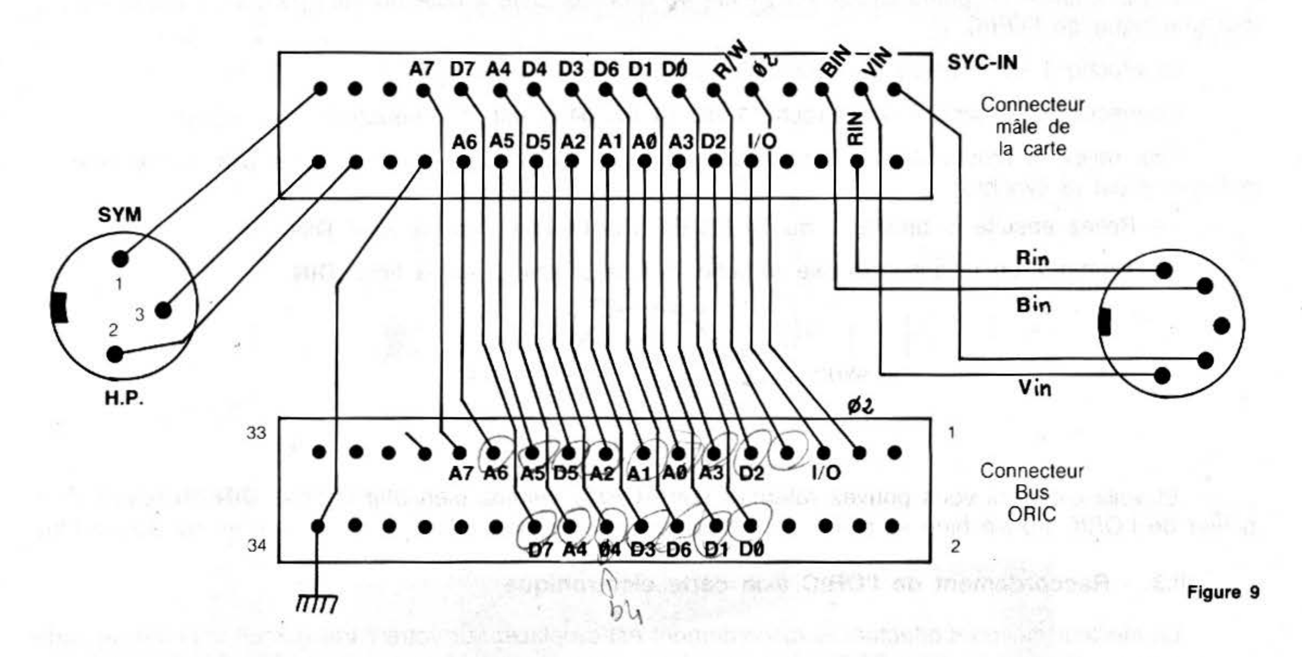


Figure 9

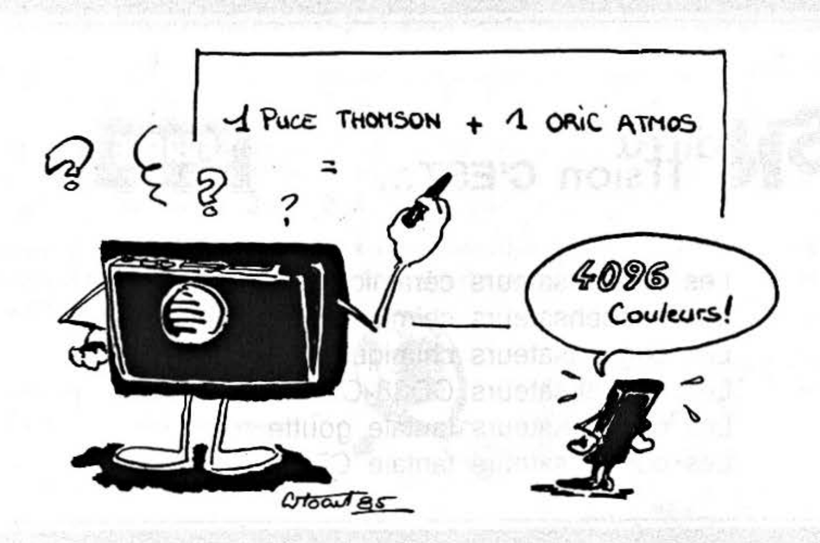
• Les 3 fils les plus à gauche doivent être séparés de la nappe et reliés à la fiche **DIN** 3 broches. Longueur 25 cm .

• Les 5 fils les plus à droite doivent aussi être séparés et reliés à la prise **DIN** 5 broches (R, V, B). Longueur 20 cm .

• 3 fils de chaque côté peuvent aussi être détachés et coupés au ras du connecteur ils ne seront pas utilisés.

• Les 20 fils restants doivent être raccourcis à 15 cm puis raccordés au deuxième connecteur femelle 34 broches.

Ici l'opération est un peu plus délicate car s'ils occupent 20 positions jointives sur le connecteur carte, il n'en n'est pas de même sur le connecteur Bus, et il faut veiller à suivre fidèlement le plan Figure 9 en écartant les fils qui ne sont pas destinés à des positions jointives, avant de sortir la nappe.



La photo n° 4 ci-dessous montre la nappe de raccordement entièrement réalisée.

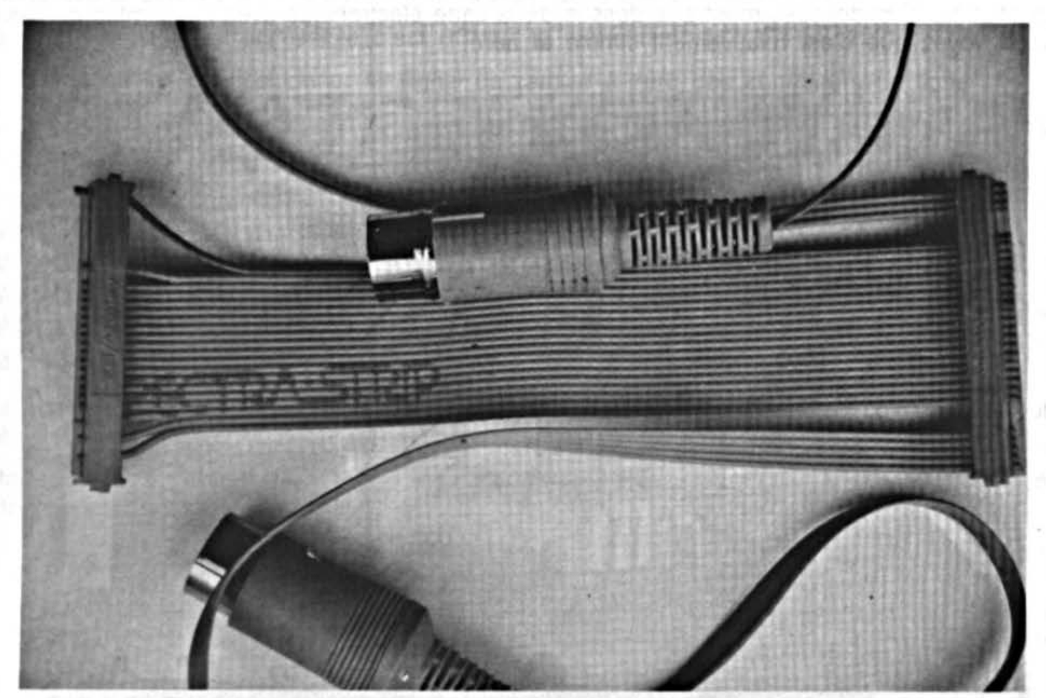
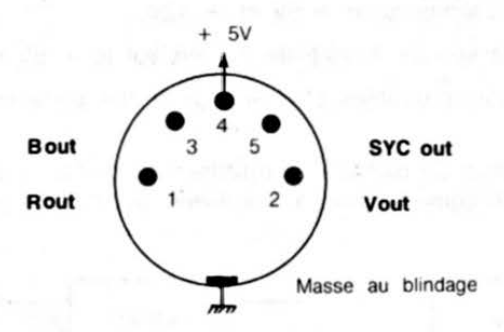


Photo n° 4

III.4. - Raccordement au téléviseur

Ce dernier raccordement est très facile, il suffit de connecter les sorties de la carte électronique sur une prise **DIN** 5 broches femelle suivant le schéma ci-dessous.



Cette prise femelle doit être fixée sur la boîte contenant votre extension.

Le câble péritel d'origine de votre *ORIC* viendra se brancher sur cette prise sans autre transformation. Mais vous pouvez même vous offrir encore un petit perfectionnement et supprimer l'alimentation Péritel extérieure.

Il vous suffit de raccorder un couple de fils (1 brin + masse) muni de la prise Jack appropriée à l'alimentation de votre câble péritel, et d'y raccorder le + 12 V disponible sur le TEA 5114. L'alimentation de votre prise viendra alors directement de la carte et votre transfo péritel ne sera plus utile.

Notons que l'inverse ne peut être réalisé car l'alimentation péritel n'est pas assez puissante pour alimenter votre carte.

Pour finir notons que certains *ORIC* importés en France ont subi une légère modification permettant de supprimer l'alimentation péritel additionnelle. Pour ceux-ci et pour ceux-ci seulement, le câble péritel a été modifié pour recevoir sur la broche 5 le + 9V non régulé disponible sur l'entrée d'alimentation de

l'*ORIC*. Il vous suffit alors de connecter le + 12V de votre carte sur la broche 5 de la fiche **DIN** femelle. Votre câble péritel est alors utilisable tel quel.

La photo n° 5 ci-dessous montre le dessus de la carte électronique avec l'ensemble de ses composants. Vous voyez que c'est relativement simple et aéré.

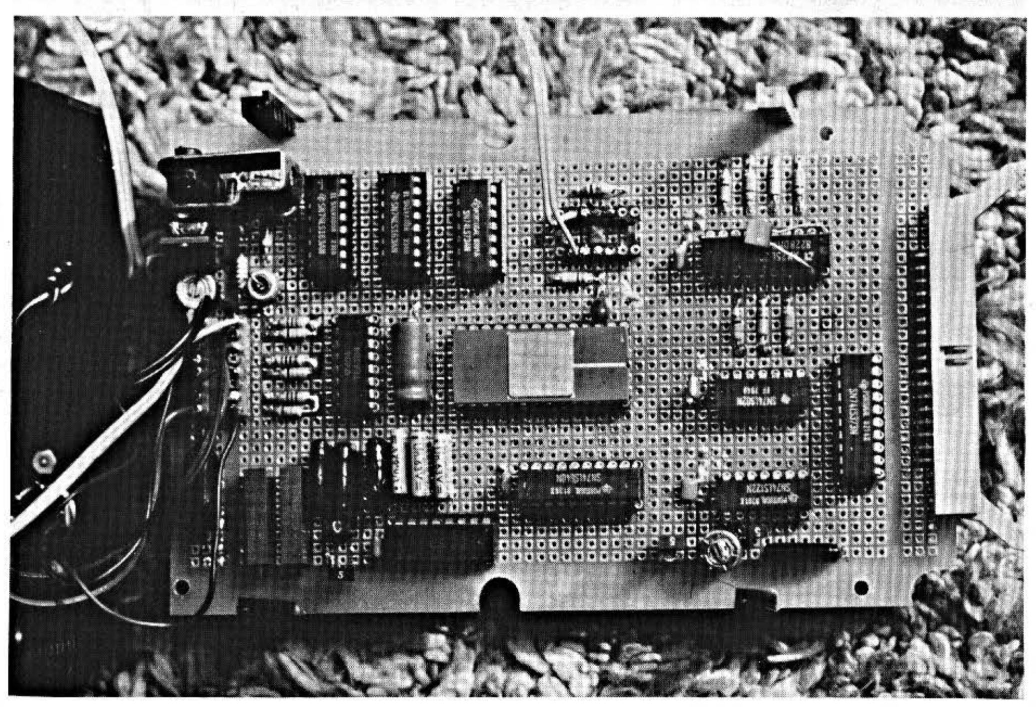


Photo n° 5 : carte électronique.

III.5. - Alimentation

Il faut deux tensions d'alimentation + 5V et + 12V.

Les courants absorbés sont de l'ordre de 50 mA sur le + 5V et 40 mA sur le + 12V.

Il existe des alimentations doubles standard qui feront parfaitement l'affaire sauf peut-être sous l'angle du prix.

Il est très simple d'utiliser un bloc standard redresseur + 12V - 500 MA. (Attention les blocs moins puissant + 12V - 150 mA ne conviendront pas au niveau puissance, ils chaufferont et le + 12V tombera à 8-9V).

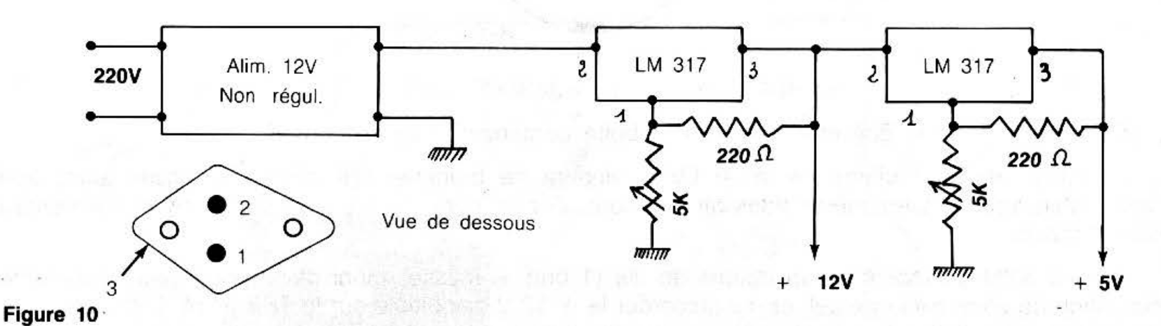


Figure 10

Nous conseillons l'emploi du LM 317 en boîtier métallique de puissance TO3 ce qui évite de mettre des radiateurs.

On peut aussi diminuer la consommation en baissant le 12V à 9-10V le TEA 5114 fonctionne encore très bien. Par contre pour le + 5V il vaut mieux régler à 5,25V.

III.6. - Derniers conseils et mise au point

Nous avons déjà vu les réglages de la fenêtre de blanking par R1 et R2.

Un autre réglage est accessible c'est R3 sur le monostable M3 (74 LS 122).

Il permet en fait de positionner le signal de remise à zéro des compteurs de ligne.

Si votre carte fonctionne à la mise sous tension, vous devez observer un écran coloré avec une séparation horizontale.

Entrez ce petit programme :

```
1Ø FOR I = 51 TO 251
2Ø POKE # 3F9,I
3Ø WAIT 1Ø
4Ø NEXT I
5Ø GOTO 1Ø
```

La séparation doit se déplacer dans les limites de la zone d'affichage. Réglez R3 jusqu'à obtention de ce résultat.

Mais si vous en êtes là, c'est que votre extension marche, sinon revoyez bien votre montage et vérifiez toutes les interconnexions.

III.6.1. - Réglage de l'horloge point H.P.

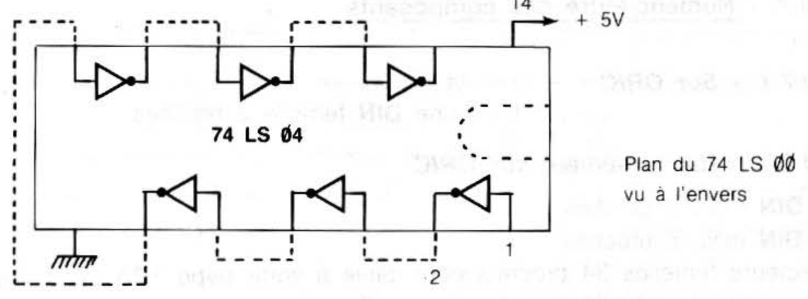
Il se peut qu'à la programmation, ou en fonctionnement certaines couleurs recopient le N° de couleur voisin. Ceci est dû à l'extrême sensibilité du EF 9369 (Il accepte un H.P. jusqu'à 17 MHz ! ! !).

Il va vous falloir ajuster la phase de H.P. 1er remède, modifiez la tension d'alimentation + 5V, de + 4,5V (pas moins) à + 5,5V (pas plus).

2ème remède : ouvrez votre *ORIC* et intercalez des inverseurs sur le 74 LS Ø4 d'adaptation de H.P. que vous avez positionné au chapitre III.2.

Nous rappelons sur la Figure 11 ci-dessous le schéma de ce circuit vu à l'envers tel que vous l'avez codé.

Figure 11



La broche 1 souvenez-vous est reliée au C.I. graphique broche 7.

La broche 2 va sur la fiche **DIN**.

Vous pouvez intercaler entre cette broche et la fiche **DIN** autant d'inverseurs qu'il faudra pour obtenir une stabilité des couleurs.

III.6.2. - Recherche de pannes

• Vous n'avez pas d'image (vérifiez en basculant le TEA 5114 sur la sortie directe *ORIC*) commutateur manuel (voir III.1.7).

Si vous n'avez toujours pas d'image votre branchement Péritel est peut être défectueux, rebranchez directement l'*ORIC*. Ça doit marcher. Vérifier alors le montage du TEA 5114 et en particulier les alimentations. L'image à travers le TEA 5114 doit revenir.

• Votre clavier est bloqué bien que vous ayez une image. (Pas de keyclick).

Débranchez le connecteur d'extension. Votre *ORIC* doit à nouveau marcher, mais il faut vérifier toutes les connexions bus.

ATTENTION NE MANIPULEZ PAS LE CONNECTEUR D'EXTENSION AVANT D'AVOIR TOUT DÉBRANCHÉ.

• Après réparation des erreurs vous devez avoir accès au clavier. Mais si vous n'avez toujours pas d'image à travers l'extension, effectuez plusieurs mises sous tension, pour le cas ou les mémoires du circuit palette ne comporteraient que des zéros ce qui provoquerait un affichage entièrement noir.

• Vous pouvez aussi essayer ce petit programme (pour l'écrire puis le sauvegarder rebranchez normalement votre *ORIC*).

```
10 POKE # 3FF,255 : POKE # 3FD. 248
2Ø WAIT 5
3Ø GOTO 1Ø
```

L'écran devrait afficher des teintes variables aléatoirement.

Si vous n'avez toujours rien, revoyez votre circuit vidéo. Suivez le schéma depuis les broches 4-5-6 du EF 9369 jusqu'à votre fiche PÉRITEL. Il y a sûrement plusieurs erreurs. Le TEA 5114 doit être un peu chaud, s'il est alimenté depuis quelques minutes, sinon c'est qu'il ne reçoit pas le 12 V.

• Voilà, ça y est vous avez une image.

Lancez le programme ci-dessus et même si l'écran ne change pas de couleur, vous devez apercevoir des petits traits noirs sur l'écran, ce sont les traces de l'accès au circuit palette par les adresses # 3FF, # 3FD.

Si vous ne voyez rien, il doit y avoir une erreur dans les circuits de décodage, vérifiez bien le câblage des adresses et aussi des codes de comparaison (broches 1-3-5-7-8-9-10-11-14-15-17-20.

• Passez maintenant au programme de démonstration fourni au chapitre IV. Il doit fonctionner, vous aurez peut-être encore de petites erreurs dans les couleurs.

Revoyez s'il n'y a pas d'inversion dans les circuits vidéo R, V, B et dans les données DØ-D7.

VOILA TOUT EST EN ORDRE.

III.7. - Nomenclature des composants

III.7.1. - Sur ORIC

1 - 74 LS ØØ
1 - Fiche **DIN** femelle 3 broches

III.7.2. - Raccordement sur ORIC

1 fiche **DIN** mâle 3 broches.
1 fiche **DIN** mâle 5 broches.
2 connecteurs femelles 34 broches pour câble à sertir (type TBA 6Ø93 4ØØ M).
1 nappe de fils à sertir 34 brins longueur 3Ø cm.

III.7.3. - Raccordement Péritel

1 fiche femelle **DIN** 5 broches.

III.7.4. - Alimentation

1 bloc alimentation + 5 V régulés 1ØØ mA
+ 12 V régulés 1ØØ mA.

ou bien

1 bloc de redressement 12 V – 5ØØ mA.
2 régulateurs LM 317 en TO3.
2 potentiomètres 5k Ω (miniature).
2 résistances 220 Ω 1/4 W.

Éventuellement 1 fiche jack pour alimentation Péritel.

III.7.5. - Carte électronique d'extension multicoloric

Composant	Ext. 8 couleurs	Ext. 16 couleurs
• Circuit imprimé ou à wrapper 105 mm × 180 mm.	1	1
• Connecteur mâle coudé 34 broches (type TBA 6Ø934Ø7ES).	1	1
Résistance 4,7 kΩ	7	8
Résistance 390 Ω	3	3
Résistance 1,2 kΩ	3	3
Résistance 1 kΩ	3	3
Résistance 68 Ω	3	3
Potentiomètre miniature 10 kΩ	2	3
Condensateur 1ØØ pF	1	1
Condensateur 1,2 nF	1	1
Condensateur 6 nF	1	1
Condensateur 4,7 nF	Ø	1
Condensateur 1ØØ nF	6	6
Condensateur tantale 1Ø µF	1	1
Condensateur 47 nF	Ø	1
C.I. 25 LS 2521	1	1
C.I. 74 LS Ø2 ou 74 HC Ø2	Ø	1
C.I. 74 LS ØØ	1	1
C.I. 74 LS 221 74 HC 221	1	1
C.I. 74 LS 122 74 HC 122	Ø	1
C.I. 74 LS 163 74 HC 163	Ø	2
C.I. 74 LS 373 74 HC 373	Ø	1
C.I. 74 LS 73 74 HC 73	Ø	1
C.I. 74 LS 64Ø 74 HC 64Ø	1	1
C.I. EF 9369 circuit palette	1	1
C.I. TEA 5114 circuit vidéo	1	1
Interrupteur manuel (modèle quelconque)	1	1
En option 1 voyant témoin (ex. 12 V - 20 mA)	1	1

Tous ces composants sont parfaitement standard à l'exception des deux derniers C.I.
— Circuit intégré palette EF 9369.
— Circuit intégré vidéo TEA 5114.

Ces deux circuits sont désormais commercialisés mais n'existent qu'au catalogue Thomson semi-conducteurs. Ils ne peuvent donc être procurés qu'auprès des distributeurs Thomson semi-conducteurs, ou éventuellement directement auprès des services commerciaux à Vélizy.

IV. - PROGRAMMATION DE MULTICOLORIC

IV.1. - Positionnement de la séparation

Les N^{os} de ligne de la zone active de l'écran généré par *ORIC* vont de 51 à 251.

Si **S** est le N° de ligne de séparation souhaité (Ø à 2ØØ en **HIRES**) faites **POKE #** 3F9,**S** + 51 et la séparation se déplace sur l'écran.

Difficile n'est-ce pas!!

Un peu plus difficile maintenant.

IV.2. - Programmation des couleurs

Sauf erreur de câblage la correspondance entre les couleurs est la suivante :

	N° ORIC	N° MULTICOLORIC
Zone Supérieure	Ø	Ø
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
Zone Inférieure	Ø	8
	1	9
	2	1Ø
	3	11
	4	12
	5	13
	6	14
	7	15
		COULEUR = C

La programmation la plus simple d'une couleur se fait en quatre temps.

- 1er temps adressage N° de couleur Ø-15.
  Placer 255 – 2 × C dans # 3FF.
- 2e temps : programmation de la dose de rouge et de bleu dans la couleur "C".
  R = Niveau de rouge de Ø à 15.
  B = Niveau de bleu de Ø à 15.
  Placer 255 – 16 × B – R dans # 3FD.
- 3ème temps : adressage N° de couleur (deuxième donnée).
  Placer 254 – 2 × C dans # 3FF.

— Les différentes nuances de gris s'obtiennent par un mélange équilibré des 3 couleurs.

Alors amusez-vous bien !

IV.3. - Programme DEMO

Le listing en est donné ci-dessous.

— Le mouvement de la séparation s'arrête en haut de l'écran après un appui sur une touche quelconque

— L'interrogation S/C apparaît :

✻ S : séparation

Le programme vous demande ensuite le N° de ligne Ø-2ØØ.

✻ C : couleur

Vous choisissez ensuite le N° de couleur à programmer Ø-15 puis les dosages R, B, V.

La barre d'espace permet de revenir en mode balayage de l'écran.

✻ Redéfinition des bords de l'écran. La couleur Ø redéfinira les bords supérieurs, la couleur 8 les bords inférieurs.

LISTING PROGRAMME DEMO

```
2 DATA #AD,#00,#04,#8D,#F9,#03,#60
5 FORI=#401 TO #407:READX:POKEI,X:NEXT
10 DATA #AD,#F0,#04,#8D,#FF,#03
20 DATA #AD,#F1,#04,#8D,#FD,#03,#60
30 FORI=#4F2 TO #4FE:READX:POKEI,X:NEXT
40 HIRES:FORI=0 TO 210 STEP 30:CURSETI,0,0:FILL200,1,16+I/30:NEXT
42 REPEAT:FORI=51TO251:POKE#400,I:CALL#401:NEXT
43 FORI=251TO51STEP-1:POKE#400,I:CALL#401:NEXT:UNTILKEY$<>""
45 INPUT"Separation ou Couleur --> S/C ";S$
46 IF S$<>"S" THEN 48
47 PRINT:INPUT"Separation 0-200";S:POKE#400,S+51:CALL#401:CLS:GOTO45
48 IF S$<>"C"THEN 42
50 CLS:INPUT"No de couleur 0-15 --> ";C:POKE#4F0,255-2*C:CLS
60 INPUT"% de Rouge 0-15 --> ";R
70 INPUT"% de Bleu  0-15 --> ";B
80 INPUT"% de Vert  0-15 --> ";V
90 POKE#4F1,255-16*B-R:CALL#4F2
100 POKE#4F0,254-2*C:POKE#4F1,255-V:CALL#4F2
110 CLS:GOTO45
```

- 4ème temps : programmation de la dose de vert dans la couleur "C".
  V = Niveau de vert de Ø à 15.
  Placer 255 − V dans # 3FD.

En réalité le circuit palette dispose d'un adressage auto-incrémental et le 3ème temps n'est pas indispensable.

Deux adressages successifs en # 3FD écrivant automatiquement le R et B puis le V.

De plus si l'on souhaite gagner du temps machine, il n'est pas nécessaire d'adresser le N° de couleur à chaque couleur, les deux adressages suivants en # 3 FD, écrivant automatiquement dans la couleur C + 1.

Mais ceci est un peu du perfectionnisme, nous recommandons la 1ère méthode, la plus sûre.

Une remarque cependant, le mode incrémental automatique fait que le circuit palette n'accepte pas les adressages par la routine *BASIC* **POKE.** En effet celle-ci, sur l'*ORIC* active plusieurs fois la même adresse lors de l'exécution de la routine **POKE** ce qui perturbe le fonctionnement du circuit palette à l'écriture.

De ce fait la programmation directe des couleurs par des **POKE** en # 3FF et # 3FD n'est pas utilisable. Il faudra donc faire une petite routine en langage machine.

En voici un exemple :

Routine machine en # 4Ø2.

FOR I = # 402 TO # 40E : Read X : POKE I, X : NEXT
avec DATA : # AD, # 00, # 04' charger 400 dans ACC
DATA : # 8D, # FF, # 03' transférer dans # 3FF
DATA : # AD, # 01, # 04' charger 401 dans ACC
DATA : # 8D, # FD, # 03' transférer dans # 3 FD
DATA : # 60' retour BASIC.

Ces routines sont équivalentes à des
POKE # 3FF, PEEK (# 400)
POKE # 3FD, PEEK (# 401)

Mais avec une seule activation des adresses # 3FF et # 3FD.

L'activation *BASIC* est la suivante :
POKE # 400, 255 − 2 × C
POKE # 401, 255 − 16 × B − R
CALL # 402
POKE # 400, 254 − 2 × C
POKE # 401, 255 − V
CALL # 402

Pour vous exercer à maîtriser l'influence des différentes composantes couleur, utilisez le ***programme de Demo*** donner plus loin et notez la combinaison R, V, B des coloris qui vous plaisent.

A titre indicatif vous pouvez reconstituer les couleurs d'origine d'*ORIC* en saturant les composantes à Ø ou à 15.

	=	R	V	B
Rouge		*15*	*0*	*0*
Bleu	=	*0*	*0*	*15*
Vert	=	*0*	*15*	*0*
Jaune	=	*15*	*15*	*0*
Cyan	=	*0*	*15*	*15*
Magenta	=	*15*	*0*	*15*
Blanc	=	*15*	*15*	*15*
Noir	=	*0*	*0*	*0*

— Vous pouvez obtenir une palette pastel en remplaçant les "Ø" par des 3 et les 15 par des 12.

— Vous pouvez obtenir des teintes dégradées d'une couleur en variant sur dosage de Ø à 15.

La photo ci-dessous représente un exemple d'écran généré par le programme Demo.

On y distingue aisément :

— La séparation (placée "ici" au milieu),

— la palette supérieure (nous avons régénéré les 8 couleurs d'origine *ORIC* avec fond d'écran noir).

— La palette inférieure avec 8 nouvelles couleurs dont le noir redéfini en gris (R = 3 ; B = 3 ; V = 3). On n'y constate que les bords d'écran, autour de la zone rectangulaire utile.

Remarquez au passage la précision du rendu des couleurs à travers l'extension et l'absence de génération de parasites, (moirages, superposition de couleurs aux transitions).

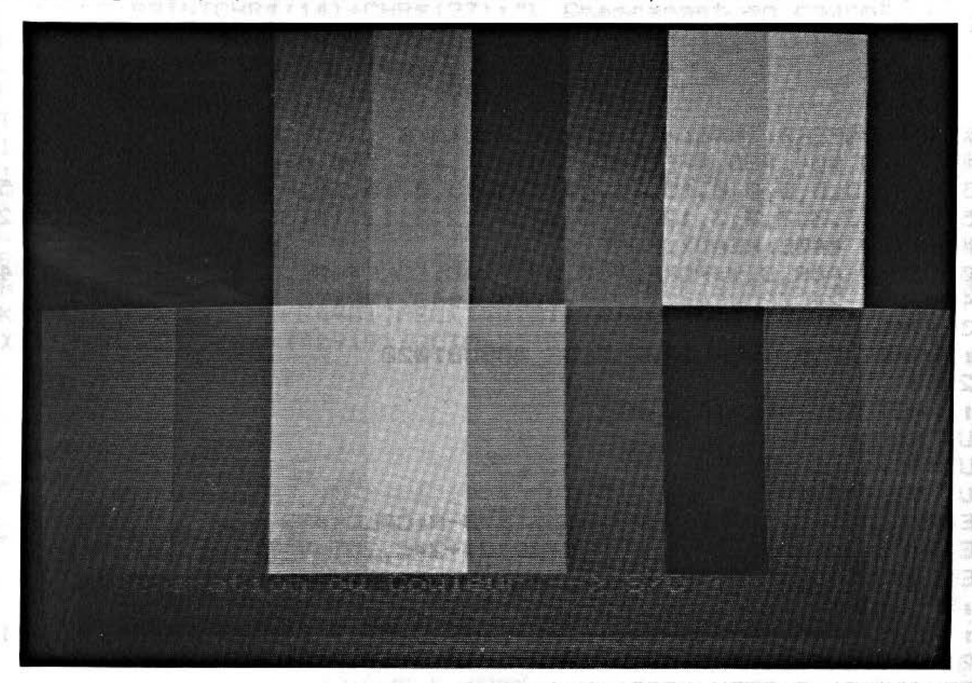


Photo N° 6
Image Demo

IV.4. - Programme de peinture

Ce programme permet de dessiner en **HIRES** en exploitant toutes les possibilités de l'extension *MULTICOLORIC.*

A la mise en route du programme un curseur apparaît en haut à gauche (position Ø,Ø). Il se déplace avec les flèches de direction sans modifier l'écran. L'utilisation de **SHIFT** + flèches accélère les déplacements verticaux.

Les commandes sont :

"S" — Séparation

"C" — Couleur

— Ces commandes fonctionnent comme dans le programme DEMO.

"F" — Écriture d'un octet à l'emplacement du curseur. Cette commande vous interroge sur la valeur à afficher. Ces valeurs ont leur rôle habituel Texte Ø-7
Fond 16-23
y compris les couleurs inversées (128 à 255). S'il y a erreur, retapez "F", sinon barre d'espace, le curseur se déplace automatiquement laissant la place à l'octet que vous venez d'écrire sur l'écran.

"R" — Écriture de N lignes × N colonnes avec la même valeur.
Même fonctionnement que "F" mais très utile pour dégrossir un dessin par grands pavés de couleur.

Nota : quand le curseur occupe une place ou se trouve déjà un attribut couleur, il perturbe la ligne sur lequel il se situe. L'attribut est restitué après le déplacement du curseur.



Les dernières commandes sont :

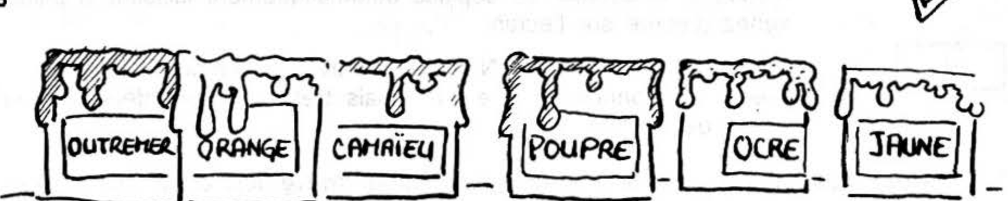
"L"	Pour lecture de cassette
"E"	Pour écriture de cassette

Ces deux commandes doivent être confirmées par l'appui d'une touche (n'oubliez pas de mettre le magnétophone en route !)

— La programmation des couleurs et de la séparation est enregistrée en même temps que votre dessin, elle est restituée à la fin de la lecture.

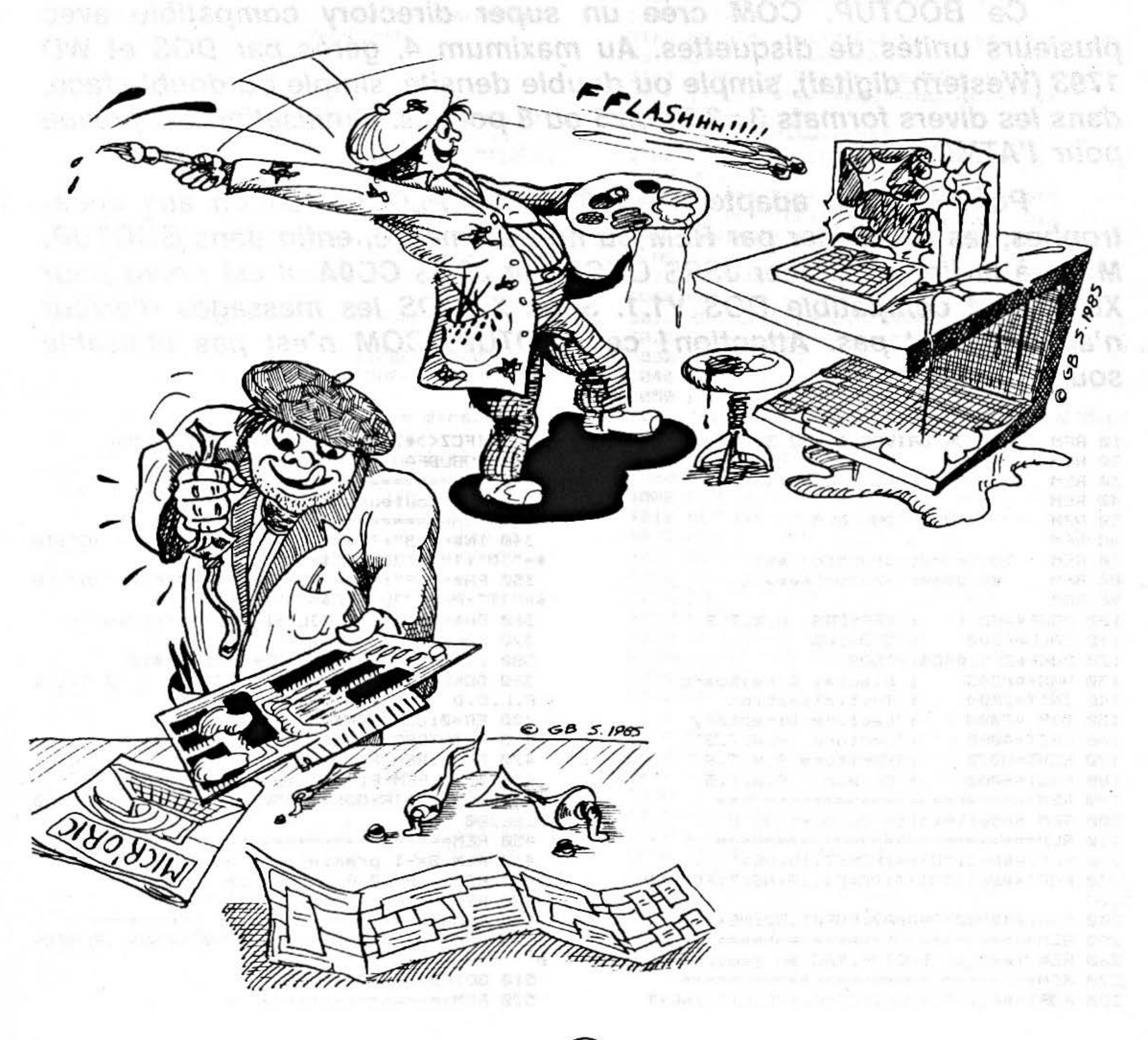
ET MAINTENANT A VOS PINCEAUX

```
10 HIMEM#97FF:PAPER0:INK6:HIRES:PAPER0:INK0
30 DATA #AD,#00,#04,#8D,#FF,#03,#AD,#01,#04,#8D,#FD,#03,#60
35 DATA13,0,3,15,0,0,0,0,15,15,0,15,0,15,0,15,15,0,0,15,15,15,15,15
36 DATA 5,8,2,12,3,3,3,3,12,12,3,12,3,12,3,12,12,3,3,12,12,12,12,12
40 FORI=#402 TO #40E:READX:POKEI,X:NEXT
45 FORC=0TO15:READR,B,V:POKE#400,055-2*C:POKE#401,255-16*B-R:CALL#402
46 POKE#400,254-2*C:POKE#401,255-V:CALL#402:NEXTC
47 POKE#3F9,151
50 GETX$:IF X$<>"C" THEN 2000 ELSE GOSUB1020
60 GOTO 2000
1020 CLS:INPUT"Couleur";C
1030 INPUT"Rouge";R
1040 INPUT"Bleu";B
1050 INPUT"Vert";V
1060 POKE#400,255-2*C:POKE#401,255-16*B-R:CALL#402
1065 POKE49120+2*C,255-16*B-R:POKE49121+2*C,255-V
1070 POKE#400,254-2*C:POKE#401,255-V:CALL#402
1090 CLS:RETURN
1100 CLS:INPUT" Separation 0 - 200 ";X:POKE#3F9,X+51:POKE40919,X+51:RETURN
2000 P=40960:GOSUB3000
2110 A=PEEK(#208):B=PEEK(#209):N=1
2111 IF A=#38 AND B=#38 THEN 2110
2112 IF B=#A4 OR B=#A7 THEN N=8
2115 IF A=#BA THEN GETX$:GOSUB1020:GOTO2110
2120 IF A=#9E THEN 3200
2125 IF A=#8F THEN 3300
2130 IF A=#AC THEN GOSUB3100:P=P-1:GOSUB3000:GOTO2110
2135 IF A=#B6 THEN GETX$:GOSUB 1100:CLS:GOTO2110
2140 IFA=#BC AND P<49119 THEN GOSUB3100:P=P+1:GOSUB3000:GOTO2110
2150 IF A=#B4 THEN GOSUB3100 ELSE 2160
2155 P=P+40*N:IF P>48959 THEN P=48959
2156 GOSUB 3000:GOTO2110
2160 IF A=#9C THEN GOSUB3100 ELSE 2165
2162 P=P-40*N:IF P<40960 THEN P=40960
2163 GOSUB 3000:GOTO2110
2165 IF A=#91 THEN 2195
2170 IF A<>#99 THEN 2110
2175 GETX$:CLS
2180 INPUT"Valeur";X:POKEP,X:GETX$:IFX$="F" THEN 2180
2190 GOSUB3000:GOTO2110
2195 GETX$:CLS
```





```
2200 INPUT"Valeur";X:INPUT"N - Lignes";N
2205 INPUT"M - Colonnes";M:FOR J=0 TO M-1
2210 FORI=0TON-1:POKEP,X:P=P+40:NEXTI:P=P-40*N+1:NEXTJ
2215 GETX$:IF X$="R" THEN P=P-M:CLS:GOTO2200
2220 P=P+40*N-M:WAIT10:GOSUB3000:GOTO2110
3000 P0=PEEK(P):POKEP,170:RETURN
3100 POKEP,P0:RETURN
3200 FORW=0TO39:POKE40920+W,PEEK(48000+W):NEXT
3210 PRINTCHR$(14);"ECRAN --> CASSETTE";CHR$(27);"L ... Pret ?":GETX$
3220 GETX$:CLS:PRINTCHR$(14)+CHR$(27);"L Chargement en cours"
3230 DOKE#5F,40919:DOKE#61,49151:POKE#63,0:POKE#64,1:POKE#67,0
3240 CALL#E6CA:CALL#E57B:CALL#E804
3250 FORW=0TO39:POKE48000+W,PEEK(40920+W):NEXT:CLS
3260 GOTO 2110
3300 PRINTCHR$(14);"CASSETTE -->ECRAN";CHR$(27);"L ... Pret ?":GETX$
3310 GETX$:CLS:PRINTCHR$(14)+CHR$(27);"L Lecture en cours"
3320 DOKE#410,DEEK(#9C):POKE#67,0:POKE#35,0
3330 CALL#E6CA:CALL#E4A8:CALL#E804:DOKE#9C,DEEK(#410)
3340 FORW=0TO39:POKE48000+W,PEEK(40920+W):NEXT:CLS
3350 FOR I=0 TO 15:POKE#400,255-2*I:POKE#401,PEEK(49120+2*I):CALL#402
3360 POKE#400,254-2*I:POKE#401,PEEK(49121+2*I):CALL#402
3370 NEXTI:POKE#3F9,PEEK(40919):GOTO 2110
```





BOOTUP. COM

par Jean-Paul LEBAS

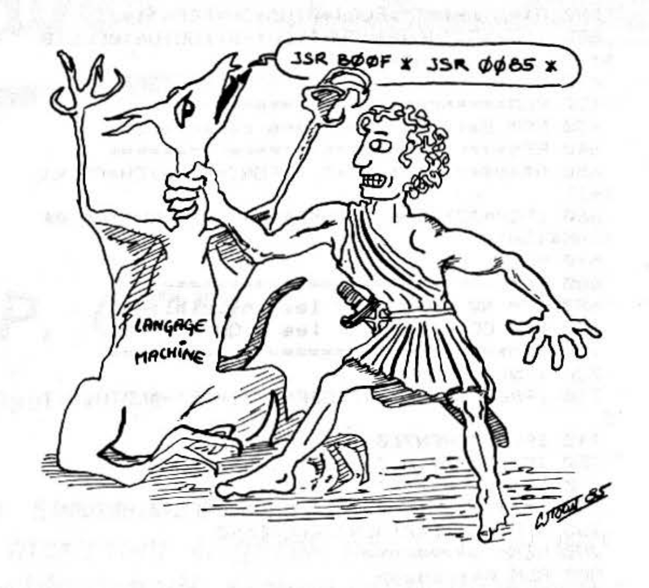
Ce BOOTUP. COM crée un super directory compatible avec plusieurs unités de disquettes. Au maximum 4, gérés par DOS et WD 1793 (Western digital), simple ou double densité, simple ou double face, dans les divers formats 3; 3,5; 5,25 ou 8 pouces. L'utilisation est prévue pour l'ATMOS.

Pour ORIC-1, adapter les PRINT AT, PLOT, attention aux apostrophes, les remplacer par REM ou ne rien mettre, enfin dans BOOTUP. MAC, à la fin, remplacer JSR$ CCCE par JSR$ CC0A. Il est prévu pour XL.DOS et compatible DOS V1.1. Sous XL.DOS les messages d'erreur n'apparaissent pas. Attention! ce BOOTUP. COM n'est pas utilisable sous RANDOS.

```
10 REM      POUR DRIVES 0 1 2 3
20 REM
30 REM      J-P  LEBAS  JUIN 1985
40 REM
50 REM         VERSION  6.5
60 REM
70 REM   Toute reproduction est
80 REM    évidement autorisee.
90 REM
100 POKE#4FD,1    :'ERREURS  R.W.T.S
110 CALL#FBD0     :'STD.CHS
120 DOKE#2F5,#4C4:'!DOS
130 HRG=#B245     :'Display & Keyboard
140 INIT=2000     :'Initialisation
150 DIR =3000     :'Lecture Directory
160 LECT=4000     :'Lecture  R.W.T.S
170 ECRI=5000     :'Ecriture R.W.T.S
180 FAUT=6000     :'Erreur   R.W.T.S
190 REM========================
200 REM Redéfinition du D et du B
210 REM========================
220 FORI=#B621TOI+4:POKEI,18:NEXT
230 FORI=#B611TOI+4:POKEI,18:NEXT:POKE#B613,28
240 FORI=#BB80TO#BBA7:POKEI,32:NEXT
250 REM===========================
260 REM Test si BOOTUP.MAC en mémoire
270 REM===========================
280 FORI=#B203TO#B261:CZ=CZ+PEEK(I):NEXT
290 IFCZ<>#2665THEN !LOAD"BOOTUP.MAC"
300 GOSUBFAUT:IFER=1THEN!DRV0:RUN
310 REM==========
320 REM Couleurs
330 REM==========
340 IN$="]@":IR$="]A":IV$="]B":IJ$="]C":IB$="]D":IM$="]E":IC$="]F":IL$="]G"
350 PN$="]P":PR$="]Q":PV$="]R":PJ$="]S":PB$="]T":PM$="]U":PC$="]V":PL$="]W"
360 DH$="]J":CL$="]L":DCL$="]N":SH$="]H"
370 CALL HRG
380 DIMN$(50),M$(50):POKE#BB81,#10
390 DOKE#12,#BB82:PRINTCL$IR$"      J.P.L & F.L.O.D  1985"IL$SH$
400 ER=0:CLS:NOM$="":GOSUBINIT
410 PRINTSPC(8)IJ$"CATALOGUE"IC$NOM$
420 DR=0:REM Flag secteur DIR
430 DT=0:REM Flag compteur noms DIR
440 GOSUB DIR:GOSUB 550:IFFIN=FALSETHEN440ELSE500
450 REM===========================
460 REM B=-1 premier nom non affiché
470 REM    B= 0 premier nom affiché
480 REM      Pratique si SYSTEM.DOS
490 REM===========================
500 IFFIN=TRUEANDDS=0ANDDP=0THENB=-1ELSEB=0
510 GOTO820
520 REM=================
```

```
 530 REM Saisies A$B$  DIR
 540 REM==================
 550 I=#B303
 560 REM===============================
 570 REM Saisie 6 premiers caractères
 580 REM===============================
 590 A$="":B$="":FORJ=0TO5:CH=PEEK(I+J)
 600 IFCH=32THENB$=CHR$(CH)+B$:GOTO610ELSEB
$=B$+CHR$(CH):A$=A$+CHR$(CH)
 610 NEXT
 620 REM===============================
 630 REM Saisie 3 derniers caractères
 640 REM===============================
 650 B$=B$+".":A$=A$+".":FORJ=6TO8:CH=PEEK(
I+J)
 660 IFCH=32THEN670ELSEB$=B$+CHR$(CH):A$=A$
+CHR$(CH)
 670 NEXT
 680 REM===============================
 690 REM NV Flag pour les invisibles
 700 REM CO Flag pour les .COM
 710 REM===============================
 720 IFNV=1THEN740
 730 IFPEEK(I+15)=#C1ORPEEK(I+15)=#C0THEN77
0
 740 IFCO=0THEN760
 750 IFMID$(B$,8,3)="COM"THEN760ELSE770
 760 N$(DT)=A$:M$(DT)=B$:DT=DT+1
 770 I=I+16:IFI>=#B3F2THENCALL#E940:RETURN
 780 IFPEEK(I)=0THEN770ELSE590
 790 REM===========
 800 REM Affichage
 810 REM===========
 820 DT=DT-1:FORJ=1TO40:B=B+1:R=R+1:IFR>7TH
ENR=1
 830 IFB>9THEN850ELSEPLOT1,J+1,R:PLOT4,J+1,
STR$(B)+"  "+M$(B):IFB=DTTHEN890
 840 IFB>8THEN860ELSEB=B+1:PLOT23,J+1,STR$(
B)+"  "+M$(B):IFB=DTTHEN890ELSENEXT
 850 PLOT1,J+1,R:PLOT3,J+1,STR$(B)+"  "+M$(
B):IFB=DTTHEN890
 860 B=B+1:PLOT22,J+1,STR$(B)+"  "+M$(B):IF
B=DTTHEN890ELSENEXT
 870 PLOT1,26,"Changer la disquette et ensu
ite RETURN"
 880 GETB$:IFASC(B$)=13THEN400ELSE880
 890 IF FIN=TRUE ANDDS<>0ANDDP<>0THEN900ELS
E910
 900 PRINT@8,25;PC$CL$IN$"Suite directory <
D>  "PN$:GOTO920
 910 PRINT@12,25;PR$CL$IN$"End of files  "
PN$
 920 PLOT1,26," Choix C - RETURN Autre face
 - ESC Fin":GETB$
 930 IFASC(B$)<>67ANDASC(B$)<>68THEN1070
 940 IFASC(B$)=68ANDFIN=TRUEANDDS<>0ANDDP<>
0THENCLS:GOTO410
 950 PLOT1,26,"-  Votre choix (No et RETURN
)  S.V.P  -":WAIT20
 960 FORI=#BF90TOI+35:POKEI,#20:NEXT
 970 PRINT@20,25;" ":FX=41
 980 IFPEEK(#208)=#97THENFX=0
 990 IFPEEK(#208)=#A8THENFX=10
 1000 IFPEEK(#208)=#B2THENFX=20
 1010 IFPEEK(#208)=#B8THENFX=30
 1020 IFPEEK(#208)=#9ATHENFX=40
 1030 IFFX=41THEN970ELSEWAIT100
 1040 WAIT50:GETFY$:FY=VAL(FY$):FZ=FX+FY:IF
INT(FZ)>44THEN970
 1050 PLOT1,26,"    -    ETES VOUS D'ACCORD
O/N ?    -    ":PRINT@19,25;INT(FZ):GETRZ$
 1060 IFRZ$="O"THEN1100ELSE950
 1070 IFASC(B$)=13THEN870ELSEIFASC(B$)=27TH
ENCLS:POKE618,3
 1080 FORI=#BB80TO#BBA7:POKEI,32:NEXT:CLS:P
APER0:INK2:!DIR:POKE#26A,3:!DRV0:END
 1090 GOTO890
 1100 A=INT(FZ):IFA<0ORA>DTTHEN950
 1110 FORI=1TODT:IFA<>ITHENNEXT:GOTO1070
 1120 FORI=#BB80TOI+40:POKEI,#20:NEXT:!LOAD
N$(A):GOSUBFAUT:RUN
 2000 '====================
 2010 ' Initialisation 2000
 2020 '====================
 2030 CLS:POKE#26A,3
 2040 FORI=0TO8:PRINT:NEXT
 2045 NN$="No Drive 0.1.2.3"
 2050 PING:PRINTSPC(9)CHR$(4)DH$IJ$NN$CHR$(
4)
 2070 PRINT:PRINT:PRINTSPC(18)" ";:GETDV$:W
AIT25
 2080 DV=ASC(DV$):IFDV<48ORDV>51THEN2030ELS
E2090
 2090 DV=DV-47:ONDVGOTO2100,2110,2120,2130
 2100 !DRV0:DV=0:GOTO2140
 2110 !DRV1:DV=1:GOTO2140
 2120 !DRV2:DV=2:GOTO2140
 2130 !DRV3:DV=3:GOTO2140
 2140 GOSUBFAUT:IFER=1THEN400
 2150 POKE#BBA3,#44:POKE#BBA4,#76:POKE#BBA5
,#3A:POKE#BBA6,DV+48:POKE#26A,2
 2155 PRINT@2,16;IM$CL$"Voulez vous tous le
s fichiers O/N":WAIT50
 2156 IFPEEK(#208)=#95THENNV=1:GOTO2165
 2157 IFPEEK(#208)=#88THENNV=0ELSE2156
 2160 GETHH$
 2162 PRINT@2,16;IM$CL$" Voulez vous tous
les (.COM) O/N":WAIT50
 2163 IFPEEK(#208)=#95THENCO=1:GOTO2165
 2164 IFPEEK(#208)=#88THENCO=0ELSE2163
 2165 GETHH$:CLS
 2170 DE=#B300:RW=#B200
 2180 DP=0:DS=1 :REM Secteur reserve
 2190 GOSUB LECT
 2200 DP=PEEK(DE+19) :REM secteur DIR
 2210 DS=PEEK(DE+18) :REM piste   DIR
 2220 FORI=DE+24TOI+8:NOM$=NOM$+CHR$(PEEK(I
)):NEXT
 2230 RETURN
 3000 REM====================
 3010 REM Lecture DIRECTORY
 3020 REM====================
 3030 GOSUB LECT
 3040 DR=DR+1
 3050 DP=PEEK(DE):DS=PEEK(DE+1)
 3060 IFDP=0ANDDS=0THENFIN=TRUE:GOTO3070ELS
EFIN=FALSE
 3070 IFDR=3THENFIN=TRUE
 3080 RETURN
 4000 REM==============================
 4010 REM Lecture Piste DP, Secteur DS
 4020 REM         256 Octets
 4030 REM==============================
 4040 POKE RW  ,DV :REM No Drive
 4050 POKE RW+1,DP :REM No Piste
 4060 POKE RW+2,DS :REM No Secteur
 4070 CALL RW+3    :REM Adr Lecture
 4080 GOSUBFAUT:IFER=1THENEND
 4090 RETURN
 5000 REM==============================
 5010 REM Ecriture Piste DP,Secteur DS
 5020 REM         256 Octets
 5030 REM==============================
 5040 POKE RW  ,DV :REM No Drive
 5050 POKE RW+1,DP :REM No Piste
 5060 POKE RW+2,DS :REM No Secteur
 5070 CALL RW+9    :REM Adr Ecriture
 5080 GOSUBFAUT:IFER=1THENEND
 5090 RETURN
 6000 REM=========================
 6010 REM Gestion erreur R.W.T.S
 6020 REM=========================
 6030 :
 6040 IFPEEK(#4FF)=4THEN6500
```

```
 6050 IFPEEK(#4FF)=6THEN6600
 6060 IFPEEK(#4FF)=1THEN6700
 6070 IFPEEK(#4FF)=0THENRETURNELSERUN
 6500 PING:CLS:PRINT@10,11;IR$DCL$"Drive no
n connecte"
 6510 PRINT@10,12;IR$DCL$"Drive non connect
e":WAIT200:ER=1:RETURN
 6600 PING:CLS:PRINT@10,11;IR$DCL$"Erreur R
.W.T.S"
 6610 PRINT@10,12;IR$DCL$"Erreur R.W.T.S":W
AIT200:ER=1:RETURN
 6700 PING:CLS:PRINT@10,11;IR$DCL$"Fichier
inexistant"
 6710 PRINT@10,12;IR$DCL$"Fichier inexistan
t":WAIT200:ER=1:RETURN
 20000 REM=================
 20010 REM Codage couleurs
 20020 REM=================
 20030 A=1281
 20040 B=DEEK(A):IFA>BTHENLIST
 20050 FORT=A+5TOB-2
 20060 IFPEEK(T)=93THENPOKET,27
 20070 IFPEEK(T)=91THENPOKET,27:POKET+1,PEE
K(T+1)-64
 20080 NEXTT:A=B:GOTO 20040
 30000 REM===================
 30010 REM Décodage couleurs
 30020 REM===================
 30030 A=1281
 30040 B=DEEK(A):IFA>BTHENLIST
 30050 FORT=A+5TOB-2
 30060 IFPEEK(T)=27THEN30080
 30070 NEXT:A=B:GOTO30040
 30080 POKET,93
 30090 IFPEEK(T+1)<32THENPOKET,91:POKE(T+1)
,PEEK(T+1)+64
 30100 GOTO30070
```



Ce programme utilise **BOOTUP. MAC** qui doit être présent sur la disquette lors de l'exécution de **BOOTUP. COM (voir ligne 29Ø)**.

Voici ce second programme, en *BASIC* et sous forme désassemblée commentée.

Chargeur MAC

```
100 FORI=#B200TO#B261
110 READD$:D=VAL("#"+D$)
120 POKEI,D:NEXT:GOTO300
130 DATA 00,00,00,4C,3B,B2,A9,21
140 DATA A0,D4,8D,85,04,8C,86,04
150 DATA A9,00,20,87,04,A2,02,BD
160 DATA 00,B2,9D,00,C0,CA,10,F7
170 DATA A9,00,A0,B3,8C,04,C0,8D
180 DATA 03,C0,20,34,B2,A9,06,20
190 DATA 87,04,58,60,BA,8E,06,C0
200 DATA 4C,90,04,A9,24,A0,D4,4C
210 DATA 0A,B2,40,EA,EA,A9,02,8D
220 DATA 6A,02,A9,01,8D,4F,02,A9
230 DATA 08,8D,4E,02,A9,03,8D,6C
240 DATA 02,A9,10,8D,6B,02,20,CE
250 DATA CC,60
300 CZ=0:FORI=#B200TO#B261:CZ=CZ
+PEEK(I):NEXT
310 IFCZ<>#2665THENPRINT"ERREUR
!!!":END
```

BOOTUP. MAC

```
B203 4C3BB2 JMP #B23B
B206 A921   LDA %#21     #D421 Vecteur écriture d'un secteur.
B208 A0D4   LDY %#D4
B20A 8D8504 STA #0485
B20D 8C8604 STY #0486    Charge en #485 et #4E6.
B210 A900   LDA %#00
B212 208704 JSR #0487    Vecteur de passage sur la RAM par #4E6.
B215 A202   LDX %#02
B217 BD00B2 LDA #B200,X  Décalage de 3 octets de #B200 en #C000.
B21A 9D00C0_STA #C000,X      1er octet = n° du lecteur.
B21D CA     DEX              2e  octet = n° de piste.
```

```
B21E 10F7   BPL #B217   3e  octet = n° de secteur.
B220 A900   LDA %#00
B222 A0B3   LDY %#B3    Préparation d'un buffer secteur en #B300 (DE).
B224 8C04C0 STY #C004   Le buffer secteur  normal est en #C023.
B227 8D03C0 STA #C003
B22A 2034B2 JSR #B234
B22D A906   LDA %#06    Vecteur passage sur la ROM par #4E6
B22F 208704 JSR #0487   en effectuant un SEI d'où le CLI qui suit.
B232 58     CLI
B233 60     RTS
B234 BA     TSX
B235 8E06C0 STX #C006   Sauvegarde pile.
B238 4C9004 JMP #0490   Exécution d'un vecteur pointé en #485-#486.
B23B A924   LDA %#24
B23D A0D4   LDY %#D4    #D424 Vecteur lecture d'un secteur.
B23F 4C0AB2 JMP #B20A
B242 40     RTI
B243 EA     NOP
B244 EA     NOP         Début de display et keyboard.
B245 A902   LDA %#02
B247 8D6A02 STA #026A   Curseur invisible et vidéo active.
B24A A901   LDA %#01
B24C 8D4F02 STA #024F   Tempo entre caractères en répétition automatique.
B24F A908   LDA %#08
B251 8D4E02 STA #024E   Tempo en répétition automatique.
B254 A903   LDA %#03
B256 8D6C02 STA #026C   Couleur encre.
B259 A910   LDA %#10
B25B 8D6B02 STA #026B   Couleur papier.
B25E 20CECC JSR #CCCE
B261 60     RTS         CLS (CC0A sur ORIC 1).
```

Quelques particularités du BOOTUP. COM

❶ Redéfinition du "D" et du "B" afin d'éviter toute confusion ave le "Ø" et le "8".

❷ Test de présence en mémoire de **BOOTUP. MAC.** Celui-ci gère les procédures RWTS (Read, Write, Track, Sector) commune à tous les **DOS,** pour les noms de fichiers dans le répertoire (Directory). Il vous est vivement conseillé de marger le **BOOTUP. MAC** au **BOOTUP. COM.** Utiliser pour cela **COPY** avec l'option, M.

❸ Saisie des 6 premiers caractères, y compris les espaces.

❹ Saisie des 3 caractères de l'extension, y compris les espaces.

❺ Saisie optionnelle des fichiers invisibles et des **. COM.**

❻ Affichage avec affectation d'un numéro.

Remarques :

Les lignes 5ØØØ-5Ø9Ø ne sont d'aucune utilité pour l'utilisation présente mais permettent l'écriture dans un secteur.

Pour d'autres utilisations, pour savoir le code d'une touche, faire, en mode direct :

FOR I = Ø **TO** 1ØØØØ : **PRINT** HEX$(PEEK(#2Ø8)) : WAIT 50 : **NEXT**

Explications de certains Flags (drapeaux) et autres variables.

ER : gestion des erreurs Ø rien, 1 erreur (I/O)
DR : compteur secteurs **DIR** explorés,
ici DR = 3 correspond à 45 noms affichés
DR = 2 correspond à 20 noms affichés
DT : compteur noms dans un secteur **DIR** exploré
DS : secteur directory
DP : piste directory
NV : flag invisibles
CO : flag **". COM"**
DV : numéro du drive
DE : buffer secteur
RW : read/write

Ligne # B3F2 = # B3ØØ + 2 + (15 × 16).

15 noms de 16 octets dont les 9 premiers sont formés du nom suivi de l'extension et dont le dernier est le statut de protection. Le nombre 2 (ajouté), correspond au secteur piste chaînage.

Pour la mise en couleur, faire **RUN** 2ØØØØ.

Pour la suppression des couleurs, faire **RUN** 3ØØØØ.

DÉROULEMENT D'UNE INITIALISATION

❶ Choix du DRIVE Ø, 1, 2, 3 : taper le n° choisi.

❷ Choix n° 1 : tous les fichiers, y compris ceux qui sont usuellement invisibles.

Choix n° 2 : affichage des seuls fichiers dont le suffixe est **". COM"**. Ceci est intéressant puisque ce sont les départs de programmes.

Pour ces deux choix, les seules réponses autorisées sont "O" ou "N". Si l'on répond "N" au premier, le choix 2 est proposé, sinon la suite du programme se déroule. Elle crée une page écran avec sur la ligne d'état, à droite le n° du drive, puis la liste des fichiers, chacun affecté d'un numéro. Enfin, en bas d'écran l'un des deux messages suivant :

❶ "END OF FILES", ce qui signifie que tous les fichiers sont affichés.

❷ "Suite directory <D>", l'appui sur la touche "D" entraînera un CLS et l'affichage de la suite.

Après cela, un nouveau choix vous est offert :

C : choix du numéro. Il faut taper 2 chiffres, après confirmation du choix le fichier correspondant est transféré.

Return : pour s'occuper d'une autre disquette ou changer de drive.

ESC : pour sortir. Affichage du répertoire automatique.

Ce programme gère les erreurs. L'affichage correspondant ne se fera que sous **D.O.S. V1.1.**

Pour certaines saisies, la méthode employée est le test de # 2Ø8 (codage touche, scrutation entrée du clavier) empêchant ainsi toute erreur.

Modification d'une chaîne par greffage rapide

par Pierre LEDAIN

L'opération réalisée par ce petit programme consiste à insérer à un endroit quelconque d'une chaîne, une autre chaîne sans modifier la longueur de la première.

Les caractères de la deuxième chaîne, qui sert de "greffe", remplacent donc ceux de la première chaîne à partir d'un rang choisi.

La longueur de la chaîne à modifier restant constante, il faut donc que celle-ci soit compatible avec la longueur de la greffe et le rang d'introduction.

Sauf omission dans le programme, si ces conditions ne sont pas réunies il ne se passera... rien.

La commande pour exécuter ce programme est simple :

— initialisez préalablement le pointeur du point d'exclamation par **DOKE** # 2F5, adresse d'implantation,

— puis ! Nom de la variable chaîne à modifier, Nom de la variable chaîne à introduire, rang d'introduction.

Ex. : ! NV$(I), M$, J

Les chaînes peuvent être indicées ou non. Les indices éventuels et le rang peuvent être un nombre, une variable ou une expression numérique.

Ex. : ! TN$(I+2), K$(3*J), N + 3 est valide.

Le programme est assez simple et je pense que les quelques commentaires données avec le listing permettront aux amateurs intéressés de le comprendre.

Pour terminer, voici un petit programme en *BASIC* pour implanter le langage machine.

Les **DATAS** en ligne 1Ø sont lues en 2ØØ et servent à l'éventuelle adaptation à l'*ATMOS*.



PROGRAMME BASIC

```
0 REM**IMPLANTATION GREFFE DE CHAINE**
1 REM** HIMEM DE PROTECTION A AJOUTER EVENTUELLEMENT********
4 DATA500.503,1051,1140,923,828,720,680,679,1134,952
6 DATA4F,03,47,06,4F,08,47,0E,0,0
10 DATAD29F,11,D188,48,D065,55
20 CLS:PRINT:PRINT"ENTREE DES OCTETS PAR PAQUETS DE 8 :":K=#400 :AD=K:P=7
30 :PRINT:PRINT"UN TEST D'ERREUR EST FAIT APRES CHAQUE PAQUET"
40 DIMSR(11):FORI=1TO11:READSR(I):NEXT:FORI=1TO11:S=0
50 FORM=KTOK+P:PRINTHEX$(M);:INPUTA$:U=VAL("#"+A$):POKEM,U:S=S+U:NEXT
60 IFSR(I)<>STHENPRINT"ERREUR,RECOMMENCEZ A:";HEX$(K):PING:S=0:GOTO50
72 K=M:NEXT
79 REM************************
80 REM* ADAPTATION ADRESSES *
81 REM************************
82 CLS:PRINT"VOULEZ-VOUS IMPLANTER LE PROGRAMME":PRINT"AILLEURS (QU'EN #400)? (O/N)"
84 GETA$:IFA$="O"THEN90
86 IFA$<>"N"THEN84ELSE95
90 PRINT:INPUT"ADRESSE D'IMPLANTATION:";AD
95 FORI=0TO#57:POKE(AD+I),PEEK(1024+I):NEXT
100 GOSUB200:IFM=0THEN130ELSEDOKE(AD+U),(AD+M):GOTO100
119 REM*********************
120 REM* ADAPTATION ATMOS *
121 REM*********************
130 IFPEEK(#D000)=166THENEND
140 FORI=1TO3:GOSUB200:DOKE(AD+U),M:NEXT:END
200 READA$:M=VAL("#"+A$):READA$:U=VAL("#"+A$):RETURN
```

Après **RUN,** tapez les octets :
#A9, #ØØ, #8D, #4F, #Ø4,.... #CF, #6Ø
avec un appui sur la touche **RETURN** après chaque octet. Les **DATAS** ligne 4 et 6 servent au contrôle de cette introduction. Ce programme *BASIC* ne sert plus à rien, ensuite si ce n'est de modèle pour implanter d'autres programmes. Il ne vous reste plus qu'à sauvegarder le langage machine de l'adresse d'implantation (AD) à AD + # 57.

Un exemple simple d'utilisation :

```
1Ø  DOKE #2F5, AD
2Ø  A$ = "ABCDEFGHI" : G$ = "XYZ"
3Ø  ! A$, G$, 4
4Ø  PRINT A$
RUN
ABCXYZGHI
```




```
0400 A900   LDA %#00      chargement en #00-01-02 du descripteur
0402 8D4F04 STA #044F     de la chaîne à modifier
0405 204704 JSR #0447
0408 A903   LDA %#03      chargement en #03-04-05 du descripteur
040A 8D4F04 STA #044F     de la chaine à introduire
040D 204704 JSR #0447
0410 200DD2 JSR #D20D     lecture du rang d'introduction
0413 A5D3   LDA #D3       protection= le rang doit être <256
0415 D02F   BNE #0446
0417 A5D4   LDA #D4       protection= le rang ne doit pas être nul
0419 F02B   BEQ #0446
041B A503   LDA #03       protection= la chaîne à introduire
041D F027   BEQ #0446     ne doit pas être vide
041F C6D4   DEC #D4
0421 18     CLC
0422 A5D4   LDA #D4
0424 6503   ADC #03       protection= vérifie la compatibilité des
0426 B01E   BCS #0446     longueurs des chaînes et du rang
0428 C500   CMP #00       d'introduction
042A F002   BEQ #042E
042C B018   BCS #0446
042E 18     CLC
042F A501   LDA #01
0431 65D4   ADC #D4       on fait pointer #01-02 sur le 1er caractère
0433 8501   STA #01       qui sera chargé dans la chaîne à modifier
0435 A502   LDA #02
0437 6900   ADC %#00
0439 8502   STA #02
043B C603   DEC #03
043D A403   LDY #03       transfert de la greffe
043F B104   LDA (#04),Y   dans la chaîne à modifier
0441 9101   STA (#01),Y
0443 88     DEY
0444 10F9   BPL #043F
0446 60     RTS           FIN
0447 20FCD0 JSR #D0FC
044A A002   LDY %#02
044C B1B6   LDA (#B6),Y   chargement du descripteur
044E 990000 STA #0000,Y
0451 88     DEY
0452 10F8   BPL #044C
0454 20D9CF JSR #CFD9     vérifie la présence de la virgule
0457 60     RTS
```

B.D.



DE RETOUR DANS LA CHAMBRE DE LUDO... BIZARRE... EN FAISANT ATTENTION ON PEUT REMARQUER UN LÉGER PAPFUM DE FAZER CHAUFFÉ A BLANC.
EH BIEN LUDO IL VA FALLOIR QUE L'ON SE QUITTE. ON REMET CA QUAND TU VEUX
JE CROS QUE POUR L'INSTANT LE SCORE SERA DUR A BATTRE... JE VAIS MÊME EN INFORMER MICR'ORIC ALLEZ SALUT B'DORIC ET BON RETOUR
APRES S'ETRE SERRE UNE CHALEUREUSE MAIS CEPENDANT VIRILE POIGNEE DE MAINS LES 2 ACCOLYTES DECIDENT DE REELEMENT SE SEPARER.
B.D'ORIC
ET DANS UNE GERBE D'ETINCELLES AUSSI EBLOUISSANTES QUE POUR SON ARRIVEE B'DORIC DISPARAIT
AH C'EST PAS VRAI IL S'EST ENCORE ENDORMI SUR CETTE MACHINE ....
HIGHT SCORE
YOU ARE THE BEST
ALLEZ LUDO... AU LIT IL EST TARD. J'ESPERE QUE TU AS TOUT DE MÊME FAIT TES DEVOIRS
OUI ROLAND IL S'EST ENCORE ENDORMI SUR CETTE MACHINE INFERNALE IL VA FALLOIR QUE TU LUI PARLES.
MAIS IL FAUT QUAND MÊME LUI ACCORDER QU'IL S'EN SERT AUSSI POUR SES LECONS
FIN
ETONNANT NON ?... ALLEZ A BIENTOT



BON DE COMMANDE MATÉRIEL

dans la limite des stocks disponibles — **Janvier 1986**

DÉSIGNATION	QUANTITÉ	P.U. T.T.C. (en francs)	P.T. T.T.C.
MATÉRIEL			
ORIC ATMOS avec son alimentation, le manuel (320 pages) en Français, une cassette de démonstration + cordon péritel. **Garanti un an**		**990**	
Protection du BUS		**30**	
Cordon Péritel		**110**	
Alimentation pour cordon péritel (12 V)		**70**	
Cordon pour moniteur monochrome		**45**	
Alimentation ORIC (9 V)		**150**	
Interrupteur ORIC		**62**	
UNITÉ A DISQUES 3 POUCES, JASMIN II (double tête, double densité, 500 Ko)		**2 800**	
MONITEUR COULEUR (Océanic)		**2 700**	
MONITEUR MONOCHROME (Goldstar)		**890**	
MODULATEUR N. et B. avec cordon		**210**	
LECTEUR K7 spécial informatique..................		**350**	
Cordon DIN, 3 jacks pour lecteur de K7		**45**	
Cordon DIN-DIN, pour lecteur d'imprimante		**990**	
IMPRIMANTE MCP 40 4 couleurs		**960**	
IMPRIMANTE GP 500 (sans cordon)..................		**2 590**	
Cordon pour imprimante GP 500		**150**	
Ruban encreur GP 500		**82**	
Rame papier (500 feuilles)		**90**	
Paire de rouleaux papier MCP 40		**40**	
Stylos pour MCP 40 (les 4 couleurs)............		**45**	
INTERFACE ROBOTIQUE (pour 2 moteurs pas à pas)		**745**	
MANETTE DE JEU (sans interface)....................		**120**	
SYNTHÉTISEUR VOCAL		**450**	
Cordon pour synthétiseur, stylo optique, carte analogique, carte 8 entrées/8 sorties		**150**	
CARTE 8 entrées/8 sorties............................		**350**	
CARTE ANALOGIQUE....................................		**350**	
CARTE MÈRE (multiprise)..............................		**270**	
LOT DE 10 CASSETTES VIERGES (10 mn)..........		**125**	
DISQUETTE VIERGE, 3 pouces		**65**	
MALLETTE DE TRANSPORT DE VOTRE MATÉRIEL (pour 1 ATMOS, 1 lecteur de K7, câbles, cassettes,...)		**220**	

TOTAL ....

Mon adresse : ____________________

____

____

____

PORT

500 F d'achat	: **30 F**
de **501 F** à **1 999 F** d'achat	: **40 F**
plus de **2 000 F** d'achat	: **80 F**

Ci-joint mon règlement par chèque.

PRIX A PAYER........

Signature :
(des PARENTS pour les Mineurs)

Adresser votre commande à :
ORIC FRANCE (Service vente aux particuliers) - B.P. 48 - 94470 BOISSY-SAINT-LÉGER

Janvier 1986

BON DE COMMANDE LIBRAIRIE ET LOGICIELS (Cassettes) dans la limite des stocks disponibles

DÉSIGNATION	QUANTITÉ	P.U. T.T.C. (en francs)	P.T. T.T.C.
LIBRAIRIE			
MICR'ORIC n°s 3 à 7 inclus + 9 (le lot)		95	
MANUEL DE L'ORIC ATMOS		45	
ORIC ET SON MICROPROCESSEUR		45	
PRATIQUE DE L'ORIC ET 36 PROGRAMMES		32	
VISA POUR L'ORIC		19	
ASSIMIL ANGLAIS (4 cassettes avec manuel)		390	
ASSIMIL ESPAGNOL (4 cassettes avec manuel)		450	
APPRENDRE LE BASIC (2 cassettes et manuel)		180	
CARTOUCHES DE JEUX MSX			
KING'S VALLEY		190	
SKY JAGUAR		190	
ROAD FIGHTER		190	
ROLLERBALL		190	
TIME PILOT		190	
CIRCUS CHARLIE		190	
MOPIRANGER		190	
YIE-AR-KUNG-FU		190	
KONAMI'S TENNIS		190	
ATHLETIC LAND		190	
CALCUL MENTAL		190	
EDDY 2		280	
MUSIC EDITOR		280	
CASSETTES DE JEUX ORIC-1 ET ATMOS			
ULTRA		70	
GREEN X TOAD		70	

DÉSIGNATION	QUANTITÉ	P.U. T.T.C. (en francs)	P.T. T.T.C.
CRIBBAGE		100	
ZEBBIE		120	
DON'T PRESS THE LETTER Q		120	
CHESS 2		120	
DAM BUSTER		120	
XENON 1		120	
ZORGON		120	
XENON 3		120	
GUBBIE		120	
PLAYGROUND 21		120	
DAMSELL IN DISTRESS		120	
LE SECRET DU TOMBEAU		140	
MISSION DELTA		180	
MR WIMPY		100	
SUPER JEEP		120	
STARTER		150	
DIAMANT DE L'ILE MAUDITE		160	
SAGA		150	
MASQUE D'OR		105	
FRELON		150	
MEURTRE A GRANDE VITESSE		180	
COBRA PINBALL		140	
DETECTIVE STORY		150	
LA FLUTE INCA		90	
UTILITAIRE			
ORIC MON		120	

Ce nouveau tarif remplace et annule le précédent

TOTAL ....

Port et emballage 0 à 500 F .... **30 F**
501 à 1 000 F .... **40 F**
1 001 à 2 000 F .... **50 F**

TOTAL A PAYER ....

Mon adresse : ______________________________

Ci-joint mon règlement par chèque.

Signature :
(des PARENTS pour les Mineurs)

Adresser votre commande à : ***ORIC FRANCE (Service vente aux particuliers) - B.P. 48 - 94470 BOISSY-SAINT-LÉGER***

SUPER INPUT

par Pierre LEDAIN

Par rapport à l'INPUT existant qui présente quelques "censures", je vous propose un INPUT de variables chaînes qui, notamment :

- ***— accepte tous les caractères du clavier ainsi que les codes ESCAPE (couleurs, clignotant...),***
- ***— peut créer une chaîne jusqu'à 254 caractères.***



1. - Commençons donc par le **détail des possibilités :**

— n'importe quel caractère du clavier peut être introduit dans la chaîne.

— en tapant ESC puis un caractère compris entre # et W, le code correspondant (couleur, clignotant) est pris en compte (cf. la table des codes ESCAPE du manuel).

— par **CTRL T,** vous entrez des majuscules ou des minuscules.

— Par **CRTL Q** le curseur apparaît ou disparaît.

— **CRTL A** recopie là où est le curseur, **y compris les codes d'affichage** (couleur, clignotant).

— **CRTL E** édite, à partir de la ligne où est le curseur, ce que vous venez d'entrer. Ceci est surtout utile après une recopie (utilisation de **CRTL A**) où l'on ne sait plus très bien comment est la mise en page.

— **CTRL G** permet d'entrer un "PING" dans la chaîne (mais il n'est pas recopiable par **CTRL A).**

— **DEL** annule un caractère à chaque fois.

— Les touches ↑⟵↑⟶ déplacent le curseur (utiles pour la recopie **(CTRL A)** où l'édition **(CTRL E)**).

— "Return" doit être tapé pour finir.

La chaîne peut recevoir 254 caractères. Si vous tentez d'en introduire plus, la chaîne sera tronquée

à 254 caractères (vous êtes avertis par "PING" de cette limite comme pour une entrée normale).

Il est intéressant avec un tel **INPUT** de passer en mode **"40 colonnes"** pour les affichages (par **POKE** # 26A, 42 (sans curseur) ou, 43 (avec curseur)).

2. - **Le programme :**

Le programme occupe 2 pages (2 fois 256 octets) : la 1re page contient le programme proprement dit et la 2e page sert de buffer (mémoire tampon).

Le listing désassemblé du programme, avec quelques indications, est fourni. Si vous êtes un peu habitué au langage machine, vous comprendrez sans grandes difficultés son fonctionnement. Il comprend trois parties principales qui sont : la détection et le traitement éventuel du caractère tapé, la mise en mémoire (dans le buffer) et en fin le traitement de la chaîne créée.

3. - **Implantation :**

Le programme peut être implanté où bon vous semble.

Il faut néanmoins :

— prévoir une place de 512 octets.

— Le protéger éventuellement par **HIMEM.**

Je vous suggère 3 emplacements possibles :

— à partir de # B2ØØ (comme sur le listing) si vous n'utilisez pas **HIRES,** avec **GRAB** et **HIMEM** B1FF possibles.

— A partir de # 96ØØ si vous voulez utiliser **HIRES,** avec **HIMEM** # 95 FF.

— A partir de # B8ØØ c'est-à-dire dans le clavier 2, si vous n'utilisez pas ce clavier.

Pour réaliser l'implantation, je vous propose le court programme *BASIC* ci-dessus :

```
0 REM**INPLANTATION INPUT.HIMEM CI-DESSOUS A ADAPTER EVENTUELLEMENT********
1 HIMEM#95FF
2 DATA1703,2157,1957,1971,2474,1875,1874,1899
4 DATA2056,2310,1655,1335,1816,2000,1554,2664
6 DATA84,1F,4B,3F,100,45,E0,56,84,70,84,7B,02,81,100,95
8 DATA84,F2,7D,F7,A7,F9,101,FB,0,0,F9,99,FA,9B,FB,D0,FC,D2
10 DATAEB78,03,EB78,0C,CCD9,48,CCD9,7E,FA9F,8F,CCB0,9D,D188,C3,D5BB,D5,CB77,D9
20 CLS:PRINT:PRINT"ENTREE DES OCTETS PAR PAQUETS DE 16:":K=#B200:AD=K
30 :PRINT:PRINT"UN TEST D'ERREUR EST FAIT APRES CHAQUE PAQUET"
40 DIMSR(16):FORI=1TO16:READSR(I):NEXT:FORI=1TO16:S=0
50 FORM=KTOK+15:PRINTHEX$(M);:INPUTA$:U=VAL("#"+A$):POKEM,U:S=S+U:NEXT
60 IFSR(I)<>STHENPRINT"ERREUR,RECOMMENCEZ A:";HEX$(K):PING:S=0:GOTO50
70 K=M:NEXT:POKEK,13
79 REM************************
80 REM* ADAPTATION ADRESSES *
81 REM************************
82 CLS:PRINT"VOULEZ-VOUS IMPLANTER LE PROGRAMME":PRINT"AILLEURS (QU'EN #B200)? (
O/N)"
84 GETA$:IFA$="O"THEN90
86 IFA$<>"N"THEN84ELSE95
90 PRINT:INPUT"ADRESSE D'IMPLANTATION:";AD
95 FORI=0TO256:POKE(AD+I),PEEK(45568+I):NEXT
100 GOSUB200:IFM=0THEN110ELSEDOKE(AD+U),(AD+M):GOTO100
110 FORI=1TO4:GOSUB200:POKE(AD+U),PEEK(AD+M):NEXT
119 REM*********************
120 REM* ADAPTATION ATMOS *
121 REM*********************
130 IFPEEK(#D000)=166THENEND
140 FORI=1TO9:GOSUB200:DOKE(AD+U),M:NEXT:END
200 READA$:M=VAL("#"+A$):READA$:U=VAL("#"+A$):RETURN
```

```
B200 A200    LDX %#00      X compte le nombre de caractères.
B202 2005E9  JSR #E905
B205 10FB    BPL #B202     lecture du clavier.
B207 C91B    CMP %#1B
B209 D01F    BNE #B22A     ESCAPE
B20B 2005E9  JSR #E905
B20E 10FB    BPL #B20B
B210 29DF    AND %#DF
B212 C940    CMP %#40
B214 90F5    BCC #B20B
B216 C958    CMP %#58
B218 B0F1    BCS #B20B
B21A 29BF    AND %#BF
B21C 0980    ORA %#80      traitement d'escape
B21E 2084B2  JSR #B284
B221 AC6902  LDY #0269     (escape doit être suivi d'un caractère
B224 9112    STA (#12),Y   entre à et W)
B226 A909    LDA %#09
B228 D053    BNE #B27D
B22A C914    CMP %#14      CTRL T
B22C F04F    BEQ #B27D
B22E C97F    CMP %#7F      DEL
B230 D008    BNE #B23A
B232 E000    CPX %#00
B234 F0CC    BEQ #B202
B236 CA      DEX
B237 18      CLC
B238 9043    BCC #B27D
B23A C905    CMP %#05      CTRL E
B23C D013    BNE #B251
B23E 8E4BB2  STX #B24B
B241 A0FF    LDY %#FF
B243 C8      INY           traitement de CTRL E
B244 B900B3  LDA #B300,Y
B247 2012CC  JSR #CC12     (affichage global de ce qui a été rentré)
B24A C000    CPY %#00
B24C D0F5    BNE #B243
B24E F0B2    BEQ #B202
B250 EA      NOP
B251 C901    CMP %#01      CTRL A
B253 D003    BNE #B258
B255 4CE0B2  JMP #B2E0     traitement de la recopie
B258 C908    CMP %#08
B25A 9006    BCC #B262     touches de déplacement du curseur
B25C C90C    CMP %#0C
B25E B002    BCS #B262
B260 D01B    BNE #B27D
B262 C907    CMP %#07      PING (mémorisé mais non recopiable)
B264 F014    BEQ #B27A
B266 C911    CMP %#11      CTRL Q
B268 F013    BEQ #B27D
B26A EA      NOP
B26B C90D    CMP %#0D      RETURN
B26D D006    BNE #B275
B26F 2084B2  JSR #B284
B272 18      CLC
B273 904D    BCC #B2C2     fin d'acquisition de caractères.Traitement de la chaîne.
B275 C920    CMP %#20      autres caractères ignorés.
B277 9089    BCC #B202
B279 EA      NOP
B27A 2084B2  JSR #B284     mise en mémoire
B27D 2012CC  JSR #CC12     affichage de caractère à l'écran
B280 4C02B2  JMP #B202     retour au début du caractère suivant
B283 EA      NOP
B284 E8      INX           mise en mémoire dans le buffer
B285 F011    BEQ #B298
B287 E0FD    CPX %#FD      a-t-on 253 caractères?
B289 9009    BCC #B294
```

```
B28B 48     PHA
B28C 8A     TXA
B28D 48     PHA
B28E 2085FA JSR #FA85          PING si 253 caractères
B291 68     PLA
B292 AA     TAX
B293 68     PLA
B294 9D00B3 STA #B300,X        stockage dans le buffer (#B301-B3FF)
B297 60     RTS
B298 A9A7   LDA %#A7           si la chaîne est trop longue on affiche
B29A A0B2   LDY %#B2           trop long.tronqué
B29C 20EDCB JSR #CBED          et on sort pour traiter la chaîne
B29F 68     PLA                (elle aura 254 caractères)
B2A0 68     PLA
B2A1 A90D   LDA %#0D
B2A3 A2FE   LDX %#FE
B2A5 D0C8   BNE #B26F
B2A7 0D 0A 09 1B 41 54 72 6F ....ATro
B2AF 70 20 6C 6F 6E 67 3B 74 p long;t
B2B7 72 6F 6E 71 75 65 07 0D ronque..
B2BF 0A 00 00
B2C2 20FCD0 JSR #D0FC          recherche de la variable
B2C5 85B8   STA #B8
B2C7 84B9   STY #B9            adresse du descripteur
B2C9 A90D   LDA %#0D           0D=RETURN- Termine la chaîne dans le buffer
B2CB 8525   STA #25
B2CD 8524   STA #24
B2CF A901   LDA %#01           adresse du 1er caractère de la chaîne
B2D1 A0B3   LDY %#B3
B2D3 EA     NOP
B2D4 2000D5 JSR #D500          traitement de la chaîne
B2D7 EA     NOP                - fait la place pour la recevoir
B2D8 202DCB JSR #CB2D          - crée un descripteur et met
B2DB A988   LDA %#88           la chaîne à l'endroit choisi
B2DD 8585   STA #85
B2DF 60     RTS
B2E0 AC6902 LDY #0269
B2E3 B112   LDA (#12),Y
B2E5 C918   CMP %#18           traitement de CTRL A
B2E7 B002   BCS #B2EB
B2E9 0980   ORA %#80           (recopie des caractères
B2EB C998   CMP %#98           y compris ceux de contrôle)
B2ED 9002   BCC #B2F1
B2EF 297F   AND %#7F
B2F1 2084B2 JSR #B284
B2F4 A909   LDA %#09
B2F6 4C7DB2 JMP #B27D
B2F9 EA     NOP
B2FA EA     NOP
B2FB EA     NOP
B2FC EA     NOP
B2FD EA     NOP
B2FE EA     NOP
B2FF EA     NOP
B300 0D4040 ORA #4040          caractère utilisé lors de l'édition
                               par CTRL E pour le retour à la ligne
```

En ligne 1 il y aura lieu d'adapter éventuellement le **HIMEM** en fonction de l'adresse d'implantation.

Après **RUN,** vous entrez les octets successifs du listing soit : A2 ØØ 2Ø Ø5 E9 1Ø FB..etc... jusqu'à EA EA, le dernier (ØD) est entré par le *BASIC*. Un test d'erreur est fait tous les 16 octets.

Ensuite vous entrez éventuellement votre adresse d'implantation (ligne 9Ø) puis le *BASIC* fait le transfert, les adaptations d'adresses et le cas échéant l'adaptation *ATMOS*.

Il ne reste plus qu'à sauvegarder la partie langage machine (**CSAVE** " ", A # adresse d'implantation, E # adresse d'implantation + 256 (idem avec disquette par **! SAVE**).

Le *BASIC* ne sert plus à rien ensuite.

Si vous implantez le programme avec un moniteur, je vous propose d'adapter éventuellement les adresses en utilisant le *BASIC* à partir de la ligne 9Ø avec les **DATA** des lignes 6 et 8 ainsi que ceux de la ligne 1Ø si vous avez un *ATMOS*.

4. - **Utilisation :**

Pour utiliser ce programme, rien de plus simple :

— vectorisez le point d'exclamation préalablement par **DOKE** # 2F5, # adresse d'implantation

— Entrez vos chaînes par **! Nom de la variable** $ indicée ou non.

Comme pour un **INPUT** normal, si la variable n'existe pas, elle est créée et si c'est un tableau supérieur à 10 termes il faut la dimensionner avant. L'indice peut être une variable numérique, ex. : ! NM$(N) affectera la chaîne entrée au N^{e} terme du tableau NM$.

Cet **INPUT** ne génère pas de point d'interrogation.



COLOREM EN L.M.

par Dany BOSC

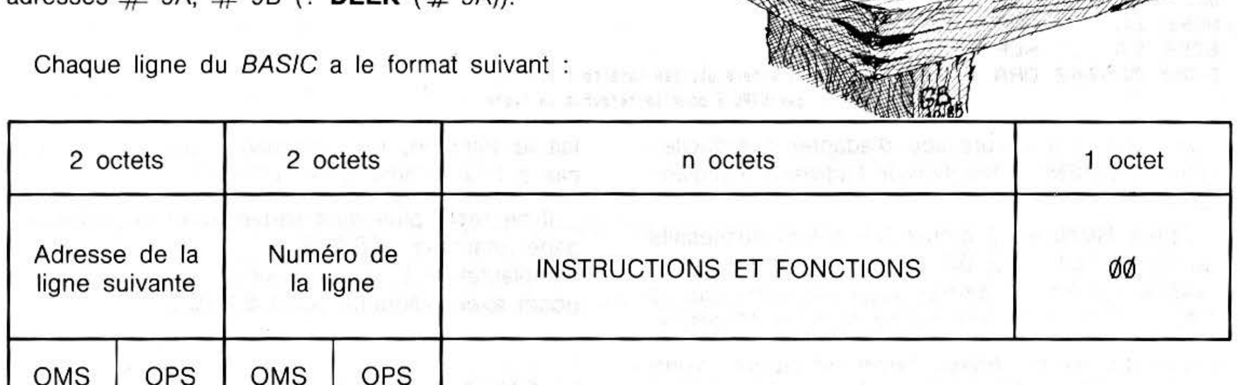
Nous revenons sur ce sujet plusieurs fois traité. Jacques François WADEL a donné un programme BASIC permettant de colorier les chaînes dans un listing BASIC. Ici il s'agit d'un programme en langage machine qui met des couleurs derrière les REMs, améliorant ainsi la lisibilité.

Le programme *BASIC* permet de charger la routine en L.M. et propose quelques exemples d'utilisation. La ligne 39 adapte le programme à l'*ATMOS*, cela revient à remplacer par Ø3 l'avant dernière donnée. Seul le module 33-38 et les **DATAS** sans **REM** sont utiles en fait, mais tout le travail fourni autour pourra aider les débutants à comprendre la mise en œuvre du procédé.

Quelques rappels permettront de mieux comprendre la genèse du programme assembleur.

L'adresse de début de programme *BASIC* est généralement # 5Ø1, elle est obtenue aux adresses # 9A, # 9B (? **DEEK** (# 9A)).

Chaque ligne du *BASIC* a le format suivant :



2 octets		2 octets		n octets	1 octet
Adresse de la ligne suivante		Numéro de la ligne		INSTRUCTIONS ET FONCTIONS	ØØ
OMS	OPS	OMS	OPS		

OMS = octet le moins significatif (octet de poids faible).

OPS = octet le plus significatif (octet de poids fort).

Les instructions et les fonctions sont stockées en un seul octet chacune. Les arguments et les noms des variables ainsi que la ponctuation sont stockées sous forme **ASCII.**

Le programme se termine par deux zéros prenant la place des 2 octets de la ligne suivante.

Forts de ces quelques renseignements, il est facile de comprendre le listing assembleur.

! et # ont été choisis pour marquer la position respective des **CHR$**(27) et **CTRL** dans les REMarques.

Libre à vous d'en choisir d'autres...

Ils sont, bien sûr, suivis par les lettres dont les effets vous sont maintenant familiers.

Exemples : **ESCAPE** A (!A) = **CHR$**(27)" A" précède et provoque l'écriture rouge.
!L caractères normaux clignotants.
!S fond jaune, etc...

CTRL L (# L) = **CHR$**(12) = code **ASCII** de L-64 provoque l'effacement de l'écran.
G fait PING
T bascule MAJUSCULES/MINUSCULES, etc...

LISTING ASSEMBLEUR COMMENTÉ

L'adresse # ØØ- # Ø1 contient l'adresse de début de ligne à tester.

L'adresse # Ø2- # Ø3 contient l'adresse de la prochaine ligne à tester.

L'adresse # Ø4 contient le drapeau **REM.**
Il est à Ø en l'absence de **REM.**
Il est à 1 après détection d'un **REM.**

B800	A59A	LDA #9A	Mettre en # ØØ- # Ø1 l'adresse de départ.
B802	A49B	LDY #9B	
B804	8500	STA #00	
B806	8401	STY #01	
B808	A000	LDY %#00	Mettre à Ø le drapeau **REM.**
B80A	8404	STY #04	
B80C	B100	LDA (#00),Y	
B80E	8502	STA #02	
B810	C8	INY	CVharger en # Ø2- # Ø3 l'adresse de la prochaine
B811	B100	LDA (#00),Y	ligne à tester.
B813	8503	STA #03	
B815	A004	LDY %#04	Incrémenter l'adresse précédente de 4 pour sauter
B817	B100	LDA (#00),Y	les octets de chaînage et de n° de ligne.
B819	C99D	CMP %#9D	→ L'octet contenu dans l'accumulateur est-il le code de
B81B	D025	BNE #B842	**REM ?**
B81D	A201	LDX %#01	NON, branchement en # B842.
B81F	8604	STX #04	OUI, mettre à 1 le drapeau **REM.**
B821	C8	INY	Charger l'octet suivant.
B822	B100	LDA (#00),Y	
B824	C921	CMP %#21	→ L'octet contenu dans l'accumulateur est-il le code
B826	D008	BNE #B830	du "!"?
B828	A91B	LDA %#1B	NON, branchement en # B830.
B82A	9100	STA (#00),Y	OUI, écrire 27 à la place de "!".
B82C	C8	INY	sauter le caractère suivant
B82D	4C21B8	JMP #B821	boucler pour un nouveau test.
B830	C940	CMP %#40	L'octet contenu dans l'accumulateur est-il le code du
B832	D00E	BNE #B842	NON, aller en # B842.
B834	A920	LDA %#20	OUI, mettre un espace (de code # 20) à la place
B836	9100	STA (#00),Y	de @
B838	C8	INY	→ Charger l'octet suivant.

```
B839 B100   LDA (#00),Y
B83B E940   SBC %#40      Lui soustraire 64 et le remplacer par la valeur calculée
B83D 9100   STA (#00),Y
B83F 4C21B8 JMP #B821     boucler pour un nouveau test.
B842 C900   CMP %#00      L'octet contenu dans l'accumulateur est-il Ø?
B844 F00A   BEQ #B850 →   OUI, aller en # B85Ø.
B846 A604   LDX #04       Sinon, charger X avec le drapeau REM.
B848 E001   CPX %#01      Est-ce 1? (Drapeau de présence de REM).
B84A F0D5   BEQ #B821 →   OUI, aller en # B821
B84C C8     INY           sinon passer à l'octet suivant
B84D 4C17B8 JMP #B817     boucler pour un nouveau test
B850 A503   LDA #03       si l'octet # Ø3 est à Ø, c'est la fin du programme
B852 C900   CMP %#00      BASIC
B854 F009   BEQ #B85F →   alors on se branche sur la routine LIST
B856 8501   STA #01       sinon on remplace l'adresse de la ligne qu'on vient
B858 A502   LDA #02       de tester par celle de la suivante
B85A 8500   STA #00
B85C 4C08B8 JMP #B808     et on boucle en tout début de ligne.
B85F 4C2CC8 JMP #C82C →   Routine LIST.
```

```
 0 REM @G@L@D
 1 REM  !NC !AO !BL !CO !DR !EE !FM@D
 2 REM
 3 :
 4 REM!L!A+---------------------+
 5 REM!L!A!                     !
 6 REM!L!A!  !H!D   Auteur      !L!A!
 7 REM!L!A! !H!E  BOSC Dany     !L!A!
 8 REM!L!A!                     !
 9 REM!L!A+---------------------+
 10 :
 11 REM  @D!J!SLANGAGE MACHINE !W@D
 12 REM
 13 REM          a implanter
 14 REM       par exemple entre
 15 REM  @D!J!S#B800  -  #B861 !W@D
 16 REM
 17 :
 18 REM!T!GMESSAGE D'ATTENTE !W
 19 :
 20 CLS:POKE #BBA3,0:POKE #26A,2
 21 M1$=" CHARGEMENT  ":M2$="   DE  LA
 ":M3$="ROUTINE L.M. "
 22 PLOT12,8,CHR$(17):PLOT26,8,CHR$(23)
 23 PLOT10,9,CHR$(3)+CHR$(10)+CHR$(17)+M
1$+CHR$(23)
 24 PLOT10,10,CHR$(3)+CHR$(10)+CHR$(17)+
M1$+CHR$(23)
 25 PLOT10,11,CHR$(3)+CHR$(10)+CHR$(17)+
M2$+CHR$(23)
 26 PLOT10,12,CHR$(3)+CHR$(10)+CHR$(17)+
M2$+CHR$(23)
 27 PLOT10,13,CHR$(3)+CHR$(10)+CHR$(17)+
M3$+CHR$(23)
 28 PLOT10,14,CHR$(3)+CHR$(10)+CHR$(17)+
M3$+CHR$(23)
 29 PLOT12,15,CHR$(17):PLOT26,15,CHR$(23
)
 30 :
 31 REM!T!GIMPLANTATION DU L.M.  !W
 32 :
 33 POKE#30E,127
 34 FOR AD=0 TO #61
 35 READ CO$
 36 CO=VAL("#"+CO$)
 37 POKE #B800+AD,CO
 38 NEXT
 39 IF PEEK(#D000)=#A9 THEN POKE #B860,3
 40 POKE #30E,192
 41 CALL#B800
 42 :
 43 REM!T!GDATAS.LISTING ASSEMBLEUR !W
 44 :
 45 DATA A5,9A    :REM!A LDA $9A
 46 DATA A4,9B    :REM!A LDY $9B
 47 DATA 85,00    :REM!A STA $00
 48 DATA 84,01    :REM!A STY $01
 49 DATA A0,00    :REM!D LDY #$00
 50 DATA 84,04    :REM!D STY $04
 51 DATA B1,00    :REM!D LDA ($00),Y
 52 DATA 85,02    :REM!D STA $02
 53 DATA C8       :REM!D INY
 54 DATA B1,00    :REM!D LDA ($00),Y
 55 DATA 85,03    :REM!D STA $03
 56 DATA A0,04    :REM!D LDY #$04
 57 DATA B1,00    :REM!D LDA ($00),Y
 58 DATA C9,9D    :REM!E CMP #$9D
 59 DATA D0,25    :REM!E BNE $B842
 60 DATA A2,01    :REM!E LDX #$01
 61 DATA 86,04    :REM!E STX $04
 62 DATA C8       :REM!E INY
 63 DATA B1,00    :REM!E LDA ($00),Y
 64 DATA C9,21    :REM!A CMP #$21
 65 DATA D0,08    :REM!A BNE $B830
 66 DATA A9,1B    :REM!A LDA #$1B
 67 DATA 91,00    :REM!A STA ($00),Y
 68 DATA C8       :REM!A INY
 69 DATA 4C,21,B8 :REM!A JMP $B821
 70 DATA C9,40    :REM!D CMP #$40
 71 DATA D0,0E    :REM!D BNE $B842
 72 DATA A9,20    :REM!D LDA #$20
 73 DATA 91,00    :REM!D STA ($00),Y
 74 DATA C8       :REM!D INY
 75 DATA B1,00    :REM!D LDA ($00),Y
 76 DATA E9,40    :REM!D SBC #$40
 77 DATA 91,00    :REM!D STA ($00),Y
 78 DATA 4C,21,B8 :REM!D JMP $B821
 79 DATA C9,00    :REM!E CMP #$00
```

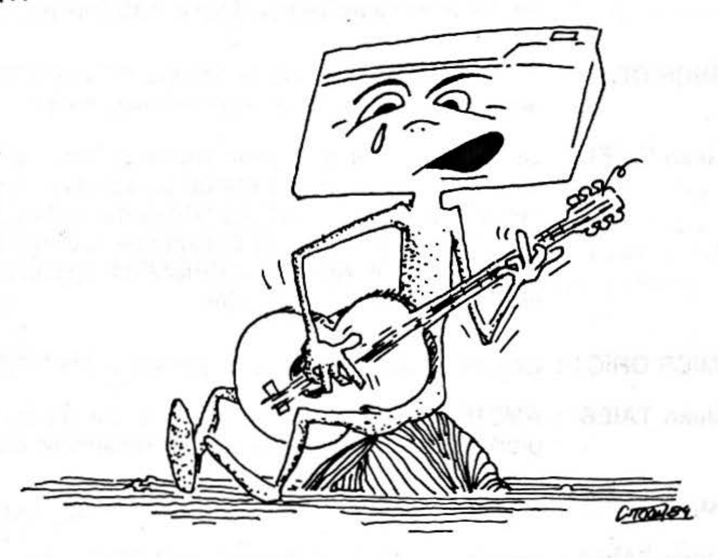
```
80 DATA F0,0A     :REM!E BEQ $B850
81 DATA A6,04     :REM!E LDX $04
82 DATA E0,01     :REM!E CPX #$01
83 DATA F0,D5     :REM!E BEQ $B821
84 DATA C8        :REM!E INY
85 DATA 4C,17,B8  :REM!E JMP $B817
86 DATA A5,03     :REM!A LDA $03
87 DATA C9,00     :REM!A CMP #$00
88 DATA F0,09     :REM!A BEQ $B85F
89 DATA 85,01     :REM!A STA $01
90 DATA A5,02     :REM!A LDA $02
91 DATA 85,00     :REM!A STA $00
92 DATA 4C,08,B8  :REM!A JMP $B808
93 DATA 4C,2C,C8  :REM!A JMP $C82C
94 :
95 REM!T!G       F I N       !W
96 :
97 REM!L!AA!DU !AR!DE!AV!DO!AI!DR@G
98 :
```




"LOTO"

Jean FINKA

```
10 REM LOTO
20 INPUT"Nombre de numeros";N
30 IF N<6 OR N>10 THEN 20
40 DIM L(49)
50 GOSUB 120
60 GOSUB 80
70 END
80 REM ===========
90 FOR I=1 TO 49
100 IF L(I) THEN PRINT I;
110 NEXT :PRINT : RETURN
120 REM ===========
130 FOR I=1 TO N
140 A=INT(RND(1)*49+1)
150 IF L(A) THEN 140
160 L(A)=TRUE :NEXT: RETURN
```





INTERVIEW DU DIRECTEUR D'A.S.N.

MICR'ORIC : La société A.S.N. Diffusion préoccupe les possesseurs de matériel *ORIC,* aussi nous vous remercions de bien vouloir faire le point.

Jean TAÏEB : Je vais m'exprimer au nom de la société A.S.N. Diffusion dont je suis actuellement le Directeur Général. Elle a 15 ans d'existence. C'est une des principales sociétés de distribution de composants électroniques professionnels sur le marché français, avec, en exclusivité les marques C.G.S., BULGIN, PHOTRONICS, RITTEL, PÉCOR...

MICR'ORIC : Quels sont vos principaux clients?

Jean TAÏEB : Nous fournissons toutes les industries électroniques.

Parmi nos principaux clients, nous pouvons citer : la TÉLÉMÉCANIQUE, THOMSON, C.G.R., C.E.A., S.A.G.E.M., S.A.F.,...

MICR'ORIC : Parlez-nous de votre expérience en micro-informatique, qui intéresse particulièrement les lecteurs de *MICR'ORIC.*

Jean TAÏEB : En 1981, la société A.S.N. a cherché de nouveaux secteurs d'activité. A cette époque, en effet, elle a du faire face à l'arrivée de gros groupes puissants sur le marché des composants. Il s'est avéré que le marché le plus porteur du moment était celui de la micro-informatique domestique. Nous avons donc ouvert notre société à ce nouveau créneau.

MICR'ORIC : Quel a été votre choix alors?

Jean TAÏEB : Au départ, nous avons commencé à commercialiser le Victor Lambda. Par la suite, début 1983, nous avons signé un contrat avec la société *ORIC U.K.* et nous nous sommes employés à commercialiser les produits *ORIC* avec le succès que l'on sait.

En 18 mois nous avons diffusé 130 000 micro-ordinateurs *ORIC-1* et *ATMOS.*

MICR'ORIC : Tout le monde sait que la société *ORIC-PRODUCT INTERNATIONAL* a déposé son bilan depuis décembre 1984. Pouvez-vous nous donner votre point de vue à ce sujet?

Jean TAÏEB : La société A.S.N. a toujours soutenu *ORIC.* Elle l'a sauvée à 2 reprises du dépôt de bilan : en août et en octobre 1984 en lui achetant massivement ses stocks. Mais, la guerre sans merci que se sont livrés les fabricants anglais a provoqué une telle chute des prix que les financiers ont été amenés à douter de la pérennité de la société. *ORIC* n'a d'ailleurs pas été la seule victime, ACORN et PRISM ont également déposé leur bilan. Quant à SINCLAIR, il a eu de très grosses difficultés.

MICR'ORIC : Comment expliquez-vous la percée d'AMSTRAD?

Jean TAÏEB : AMSTRAD a eu la chance d'arriver sur un marché laissé vide par *ORIC* et avec de bons produits, fabriqués à des prix extrêmement compétitifs en Corée.

MICR'ORIC : Quelle est votre stratégie, en matière de micro-informatique pour les mois à venir?

Jean TAÏEB : Depuis la fermeture des usines d'*ORIC,* nous continuons à écouler nos stocks à des prix



sans concurrence. Nous nous sommes tournés vers le marché du MSX qui tend à être le standard universel. Compte tenu de nos performances commerciales, la puissante société GOLDSTAR, nous a choisi pour diffuser en exclusivité sa gamme de produits : le 64 K MSX FC 2ØØ, tout d'abord. Nous commençons à commercialiser les imprimantes 4 couleurs PRT 5 et les lecteurs/enregistreurs de disquettes (au format 5,25 pouces) FD 101 (360 Ko) et FD 102 (2 fois 5ØØ Ko). C'est du matériel de qualité professionnelle.

MICR'ORIC : Et pour les utilisateurs d'*ORIC* ?

Jean TAÏEB : Compte tenu du rôle important qu'A.S.N. a joué pour la diffusion des produits *ORIC*, nous avons à cœur d'assurer nos responsabilités vis-à-vis de nos clients, comme nous l'avons toujours fait.

Pour cela, nous tenons à leur disposition tous les périphériques que nous avons créés ainsi que les produits consommables divers dont ils peuvent avoir besoin. Nous avons un stock de pièces détachées pour le service après vente.

MICR'ORIC : Comment est organisé le service après vente ?

Jean TAÏEB : Le marché de la micro-informatique ayant été explosif, nous avons mis un certain temps à mettre en place une structure de S.A.V. fiable. Actuellement, nous sommes en mesure de **dépanner sur le champ TOUT** possesseur d'*ORIC*.

MICR'ORIC : Comment cela ?

Jean TAÏEB : Nous avons crée 4 centres pilotes de dépannage sur le site :

A.S.N. Z.I. La Haie Griselle - 94470 BOISSY-SAINT-LÉGER
A.S.N. 20, rue Vitalis - 13005 MARSEILLE
A.M.I.E. 11, boulevard Voltaire - 75011 PARIS
E.F.E. 143, rue de Clignancourt - 75018 PARIS

D'autres centres pilotes sont en cours de création.

Tous ces centres serviront nos clients passés et à venir avec fidélité et efficacité.

MICR'ORIC : On lit parfois des attaques virulentes contre A.S.N. dans **THÉORIC** ou **HEBDOGICIEL.** Quels sont vos commentaires ?

Jean TAÏEB : Nous agissons avec sérénité. Nous n'avons jamais attaqué qui que ce soit. Nous constatons que ces attaques visent plutôt notre compétence, notre savoir faire et notre maîtrise du marché.

Notre souci a toujours été de satisfaire le client et c'est notre succès qui a engendré autant de commentaires. Les difficultés que nous avons parfois rencontrées soulignent le succès de l'*ORIC*.

MICR'ORIC : Souhaitez-vous que cessent ces attaques ?

Jean TAÏEB : Oui, car nous sommes pour la modération.

Nous souhaitons que nos concurrents en micro-informatique et nos collègues puissent prospérer et travailler autant qu'A.S.N. Il est amusant de constater cependant que, chaque fois que ces journaux parlent de nous, et de façon virulente, nos ventes augmentent d'une proportion considérable et nous avons alors du mal à faire face à cette subite demande.

MICR'ORIC : Quelle sera votre conclusion ?

Jean TAÏEB : L'avenir nous appartient.



RETICOL'ORIC

par Stéphane CALLÉGARI

L'objet de cet article est de décrire un algorithme de coloriage d'un contour fermé. Cet article est composé de quatre parties :

— 1 description de la mémoire image d'un micro-ordinateur.
— 2 Description de l'algorithme.
— 3 Le programme en BASIC.
— 4 Le programme en langage machine.

I. - La mémoire image

Sur *ORIC,* comme sur tous les micro-ordinateurs familiaux, les dessins apparaissant sur l'écran sont stockés, point par point, dans la mémoire vive. *ORIC,* a une mémoire image qui occupe 8 Ko entre l'adresse # ACØØ (hexa) et # BFEØ (hexa). Cette mémoire image est découpée à l'écran en 200 lignes de 40 octets. Puisque la résolution annoncée sur *ORIC* est de 200 lignes × 240 colonnes, chaque octet horizontal représente à l'écran 6 points ou pixels.

Pour mieux comprendre, effectuons ce petit test en mode direct :

HIRES: AD = # AFBØ: CO = 45: **POKE** AD,CO

Dans la suite de l'article les commandes graphiques utilisées appartiennent au *BASIC* de l'*ORIC.* (Voir liste page 4).

POKE est une commande *BASIC* permettant d'installer la valeur CO l'adresse AD. Puisque cette adresse est dans la mémoire image et que CO en binaire est égal à ØØ1Ø11Ø1 nous voyons apparaître sur l'écran le motif : – – – – .

Il faut savoir que seuls les 6 bits de droite sont le motif du dessin. Reprenons le code CO en binaire :

numéros des bits	7 6 5 4 3 2 1 Ø
l'octet	Ø Ø 1 Ø 1 1 Ø 1

A quoi servent les bits 6 et 7? Essayons :

CO = 173: **POKE** AD, CO

numéros des bits	7 6 5 4 3 2 1 Ø
l'octet	1 Ø 1 Ø 1 1 Ø 1

Nous obtenons – – , c'est-à-dire le motif inverse du précédent car nous sommes en inversion vidéo et les zéros sont des uns et inversement. Donc lorsque le bit 7 est un 1, le dessin est affiché en inversion vidéo et lorsque le bit 7 est à Ø le dessin est affiché en mode normal.

Et le bit 6? Essayons :

CO = 109: **POKE**AD,CO

numéros des bits	7 6 5 4 3 2 1 Ø
l'octet	Ø 1 1 Ø 1 1 Ø 1

Nous obtenons le dessin du 1er cas. Pas de différence?

Essayons ceci : CURSET 12Ø,1ØØ,1: CIRCLE 2Ø,1

Le cercle dessiné est inscrit sur notre dessin : – – – – .

En positionnant le bit 6 à 1 notre dessin n'est pas protégé.

Une preuve : **FOR I** = 4Ø96Ø **TO** 48959 : **POKE I,Ø** : **NEXT**

Essayez de dessiner maintenant !

Ces remarques sont naturellement faites pour *ORIC-1* ou *ATMOS.*

L'essentiel est de retenir que tout ce qui est inscrit sur l'écran est stocké en mémoire image.

Une autre possibilité de l'*ORIC* est la dépose d'attributs en mémoire. Essayons :

HIRES: CURSET 12Ø,1ØØ,1: **CIRCLE** 2Ø,1: CO = 2Ø: **POKE**AD, CO

numéros des bits	7 6 5 4 3 2 1 Ø
l'octet	Ø Ø Ø 1 Ø 1 Ø Ø

Les bits 7 et 6 sont à zéros. Le bit 5 étant lui aussi à zéro nous ne sommes pas en présence d'un dessin. Nous sommes en présence d'un attribut qui commande la couleur du papier en bleu jusqu'au bord droit de l'écran. Un **POKE** AD+8,16 vous permet de suspendre l'effet du premier attribut et de commander la couleur papier noir. Encore un petit essai :

HIRES: FOR I = 4Ø96Ø **TO** 4892Ø **STEP** 4Ø: **POKE I**,2Ø: **NEXT: CURSET** 12Ø,1ØØ,1 : **CIRCLE** 5Ø,1

Il y a dépôt de l'attribut de couleur papier bleu sur tous les octets situés sur le bord gauche de l'écran. Puisque l'effet de cet attribut est sensible sur toute la droite, tout l'écran devient BLEU ! Modifions maintenant la couleur de l'encre :

FOR I = 4Ø961 **TO** 48921 **STEP** 4Ø : **POKE I**,3 : **NEXT**

L'attribut 3 demande la couleur encre jaune. Il est déposé sur tous les 2ᵉ octets du bord gauche. Il n'y a pas de conflit entre les attributs couleur papier et couleur encre car ils sont de natures différentes.

Les commandes graphiques utilisées agissent directement sur le contenu de la mémoire image. En voici la liste :

a) **HIRES :** demande une mémoire écran de 8 Ko découpée en 200 lignes de 40 octets chacune.
b) **CURSET** X, Y, FD : allume à l'écran le point situé en (X, Y) avec la couleur du fond, si FD = Ø, ou la couleur de l'encre si FD = 1. Sur l'octet correspondant cette commande met à 1, ou à Ø, le bit concerné.
c) **CURMOV** DX, DY, FD : déplace le dernier point de DX colonnes, et de DY lignes.
d) **DRAW** DX, DY, FD : déplace le dernier point de DX colonnes, et de DY lignes en traçant un trait selon le paramètre FD.
e) **POINT** (X, Y) : est égale à VRAI si le point situé en (X, Y) est allumé, et à FAUX sinon. Note : sur *ORIC*, VRAI = – 1 et FAUX = Ø.
f) **CIRCLE** R, FD : trace un cercle de rayon R et dont le centre est le dernier point allumé.
g) **CHAR** ASC, C, FD : trace le caractère de code **ASC** et selon le mode choisi. Si C = 1 c'est le caractère graphique qui est affiché, si C = Ø c'est la lettre dont le code **ASCII** = **ASC.**
h) **FILL** L, C, CO : installe sur L lignes et C octets par ligne l'attribut de code CO.
i) **POKE** AD, CO : installe le code CO à l'adresse AD.
j) **PEEK** (AD) : récupère le code situé à l'adresse AD.

II. - Description de l'algorithme

L'algorithme décrit suppose qu'un contour fermé soit dessiné à l'écran et que l'on connaisse un point (XR, YR) situé à l'intérieur du contour. De plus, les commandes **POINT** et **CURSET** décrites sont utilisées et disponibles.

A partir du point connu, nous allons examiner à gauche puis à droite en coloriant tous les points rencontrés, jusqu'à ce que nous trouvions un point du contour ou que nous ayions atteint la limite de la mémoire image. Ensuite, il s'agit de voir ce qui se passe sur la ligne du dessus et sur la ligne du dessous, afin de dénicher les différents recoins.

Exemple : ceci est un recoin

Si nous en trouvons un, il faut empiler les coordonnées du 1ᵉʳ point du recoin afin de colorier celui-ci par la suite.

L'algorithme se déroule jusqu'à ce qu'il n'y ait plus de point empilé.

Voici l'algorithme :

```
DÉBUT     X:=XR;Y:=YR;
          EMPILER(X,Y);
          TANT QUE PILE NON VIDE FAIRE
          DÉBUT DÉPILER(X,Y);
                SI NON POINT(X,Y)
                ALORS DÉBUT A DROITE;
                            A GAUCHE;
                            DESSOUS;
                            DESSUS;
                      FIN DU SI;
          FIN DU TANT QUE;
FIN DU PROGRAMME.
```

```
A DROITE;
DÉBUT  XS:=X;
       TANT QUE X<=MAX ET NON POINT (X,Y) FAIRE
                DÉBUT CURSET X,Y,1;
                      X:=X+1;
                FIN DU TANT QUE;
        XD:=X−1;X:−XS;
FIN DE A DROITE;


A GAUCHE;


DÉBUT  X:=X−1;
       TANT QUE X>=XMIN ET NON POINT (X,Y) FAIRE
                DÉBUT CURSET X,Y,1;
                      X:=X−1;
                FIN TANT QUE;
       XG:=X+1;X:=XS;
FIN DE A GAUCHE;


DESSOUS;
DÉBUT  SI  Y<YMAX  ALORS  DÉBUT  XS:=X; YS:=Y;X:=XG;Y:=Y+1;
                                 TRONC COMMUN;
                           FIN DU SI;

FIN DE DESSOUS;


DESSUS;
DÉBUT  SI Y>Ø ALORS DÉBUT XG:=X;YS:=Y; X:=XG;Y:=Y−1;
                           TRONC COMMUN;

                      FIN DU SI;

FIN DE DESSUS;


TRONC COMMUN;
DÉBUT  TANT QUE X<=XD FAIRE
       DÉBUT  TANT QUE X<=XD ET POINT (X,Y) FAIRE X:=X+1;
              SI  X<=XD  ALORS  DÉBUT EMPILER(X,Y);
                                      TANT QUE X<=XD ET NON POINT(X,Y)
                                      FAIRE X:=X+1;
                                FIN DU SI;

       FIN DU TANT QUE;
       X:=XS;Y:=YS;
FIN DU TRONC COMMUN;
```

Il est habituel lorsque l'on décrit un algorithme de le programmer dans un pseudo-langage de programmation assez proche du *PASCAL*. Après c'est à nous de le traduire dans le langage que nous utilisons. Écrivons-le en *BASIC* dans un premier temps.

III. - Le programme en *BASIC*

Voici ce que l'algorithme donne écrit en *BASIC* :

```
1 IFXR<12ORXR>239ORYR<ØORYR>199THENEND
1Ø DIM TP(1ØØ):PT=Ø
2Ø TP(PT)=XR:PT=PT+1:TP(PT)=YR
```

```
30 FD=1:X=XR:Y=YR
40 IFPT=-1THENEND
50 Y=TP(PT):PT=PT-1:X=TP(PT):PT=PT-1
60 IFPOINT(X,Y)THEN40
70 XS=X
80 IFX>239THEN110
90 IFPOINT(X,Y)THEN110
100 CURSETX,Y,FD:X=X+1:GOTO80
110 XD=X-1:X=XS
120 X=X-1
130 IFX<12THEN160
140 IFPOINT(X,Y)THEN160
150 CURSETX,Y,FD:GOTO120
160 XG=X+1:X=XS
170 IFY=0THEN190
180 XS=X:YS=Y:X=XG:Y=Y-1:GOSUB300
190 IFY=199THEN40
200 XS=X:YS=Y:X=XG:Y=Y+1:GOSUB300
210 GOTO40
290 REM Le ss-prog TRONC-COMMUN
300 IFX>XDTHEN390
310 IFX>XDTHEN390
320 IFPOINT(X,Y)=FALSETHEN350
330 X=X+1:GOTO310
340 IFX>XDTHEN390
350 TP(PT+1)=X:PT=PT+2:TP(PT)=Y
360 IFX>XDTHEN390
370 IFPOINT(X,Y)THEN330
380 X=X+1:GOTO360
390 X=XS:Y=YS:RETURN
```

Bien sûr ceci est illisible mais le programme fonctionne.

Nous sommes très loin de l'évidente clarté de l'algorithme !

1re remarque : Le TANT QUE <condition> FAIRE DÉBUT <traitement> FIN DU TANT QUE
est remplacé en *BASIC* par :
<ligne 1> **IF NOT** <condition> **THEN** <ligne 4>
<ligne 2> <traitement>
<ligne 3> **GOTO** <ligne 1>
<ligne 4> <suite du programme>

1e remarque : Le SI <condition> ALORS DÉBUT
<traitement>;
<traitement>;
<traitement>;
FIN DU SI;
est remplacé en *BASIC* par :
<ligne 1> **IF NOT** <condition> **THEN** <ligne 3>
<ligne 2> <traitement>
<ligne 3> <suite du programme>

Testons le programme *BASIC* sur le remplissage d'un cercle troué.

Écrivons ces quelques lignes à lasuite du programme.

```
400 HIRES:CURSET120,100,0:CIRCLE50,1
410 CURSET100,90,1:DRAW50,10,1
420 DRAW-10,40,1:DRAW-10,-20,1
430 DRAW-30,0,1:DRAW0,-30,1
440 XR=120:YR=80:GOTO1
```

Tapons **RUN** 4ØØ et déclenchons le chronomètre.

Temps de coloriage : 7 minutes et 15 secondes.

Pas terrible ! L'*ORIC* est vraiment poussif de temps en temps.

IV. - Le programme en langage machine

1. - CURSET et POINT

Essayons d'améliorer les performances par un programme écrit en langage machine. Nous avons vu dans l'algorithme l'utilité des 2 commandes graphiques de l'*ORIC*, **POINT** et **CURSET.** Pour traduire l'algorithme en langage machine, il faut que l'on puisse utiliser ces deux instructions dans notre programme. Cette possibilité est expliquée dans le manuel de l'*ORIC ATMOS* mais reprenons les explications plus en détail.

N.B. : Dans la suite de l'article toutes les adresses sont données en hexadécimal et celles placées entre parenthèses concernent l'*ORIC-1.*

La fonction **CURSET** est implantée à partir de l'adresse # FOC8 (# FO2D), c'est-à-dire quelle est directement accessible par **CALL** suivi de l'adresse. Les paramètres X,Y et FD doivent être installés à partir de l'adresse # Ø2EØ(#Ø2EØ) comme ceci :

#Ø2EØ	booléen de validité des paramètres.
#02E1	valeur du paramètre X codée sur 16 bits.
#Ø2E3	valeur du paramètre Y codée sur 16 bits.
#Ø2E5	valeur du paramètre FD codée sur 16 bits.

Effectuons un petit test en mode direct :

DOKE #Ø2E1,1ØØ (valeur X = 1ØØ)
DOKE #Ø2E3,1ØØ (valeur Y = 1ØØ)
DOKE #Ø2E5,1 (valeur FD = 1)
HIRES
CALL #FOC8 (pour *ATMOS*)
CALL #FO2D (pour *ORIC-1*)

Et vous devez voir sur l'écran un petit point allumé sauf si vous n'avez ni un *ORIC-1* ni un *ATMOS* ou si vous avez oublié de brancher votre téléviseur.

La fonction **POINT** est implantée à partir de l'adresse # F1C8 (# F141). Comme pour **CURSET,** les paramètres X et Y sont installés aux adresses # Ø2E1 et # Ø2E3, le paramètre FD n'intervenant pas pour la commande POINT. Au retour de celle-ci, nous aurons à l'adresse # Ø2E1, la valeur # FF si le point (X, Y) était allumé et la valeur # Ø dans le cas contraire.

Effectuons un petit test en mode direct :

DOKE # Ø2E1,1ØØ (valeur X = 1ØØ)
DOKE # Ø2E3,1ØØ (valeur Y = 1ØØ)
HIRES
CALL # F1C8 (pour *ATMOS)*
CALL # F141 (pour *ORIC-1)*
PRINT HEX$ (**DEEK** (# Ø2E1));

Le résultat affiché est # Ø car après la commande **HIRES,** aucun point n'est allumé. Recommancez le test en tapant en 4e ligne **CURSET** 1ØØ,1ØØ,1. Le résultat est # FF car le point est allumé.

2. - Description du programme en langage machine

Le langage machine manipule les données par leurs adresses. Il n'existe pas de noms de variables comme en *BASIC,* mais cela vous le saviez déjà si vous avez lu les articles d'initiation parus dans la revue.

Pour traduire l'algorithme nous avons besoin des adresses pour XR, YR, XS, YS, XD, XG, X, Y ainsi que d'une pile et d'un pointeur sur cette pile. Voici ce qui a été décidé :

nom de la variable	adresse en mémoire
XR	# ØØØ5
YR	# ØØØ6
XS	# ØØØ2
YS	# ØØØ3
XD	# ØØØ8
XG	# ØØØ9
X	# ØØØØ
Y	# ØØØ1
PT pointeur	# ØØØ4
base de pile	# ØØØA

N.B. : A l'adresse#ØØØA (et#ØØØB) se trouve, codée sur 16 bits, l'adresse de la base de la pile (#Ø4ØØ).

Le programme est implanté à partir de l'adresse 9E82 et un grand nombre de commentaires ainsi que la correspondance avec les lignes du programme *BASIC* se trouvent sur le listing.

Reprenons le test de rapidité en tapant le programme de la page 15 et les **DATA** de la ligne 16. Déclenchons le chronomètre lorsque le dessin apparaît.

Temps de coloriage : 35 secondes.

Voilà qui est mieux mais qui est loin de rivaliser avec les temps d'un MACINTOSH par exemple. Il est possible, en améliorant l'algorithme, de diminuer encore le temps de coloriage mais je vous laisse le faire, à vous de jouer !




```
5 REM Petite démonstration
10 HIRES:GOSUB5000
20 CURSET 120,100,1:CIRCLE 50,1
30 CURSET 100,90,1:DRAW 50,10,1
40 DRAW -10,40,1:DRAW -10,-20,1
50 DRAW -30,0,1:DRAW 0,-30,1
60 POKE #05,120:POKE #06,80
70 CALL#9E82:END
```

```
4996 REM
4997 REM  Chargement du programme
4998 REM  écrit en langage machine
4999 REM
5000 REM
5010 :   READ AD$:AD=VAL(AD$)
5020 :   REPEAT
5030 :     READ CO$:CO=VAL("#"+CO$)
5040 :     POKE AD,CO
5050 :     AD=AD+1
5060 :   UNTIL CO$="EE"
5070 REM
5080 IF PEEK(#FFFE)=68 THEN 5180
5090 DOKE #9C05,#E905
5100 DOKE #9E99,#D2A0
5110 DOKE #9EDD,#F141
5120 DOKE #9EFE,#F141
5130 DOKE #9F10,#F02D
5140 DOKE #9F2D,#F141
5150 DOKE #9F3F,#F02D
5160 DOKE #9F98,#F141
5170 DOKE #9FD0,#F141
5180 RETURN
```

```
5398 REM Data des codes du coloriage
5400 DATA #9E82
5405 DATA A5,05,C9,F0,B0,10,C9,0C,90,0C,A5,06,C9,00,90,06,C9,C8,B0
5410 DATA 02,90,04,20,36,D3,60,A9,04,85,0B,A9,00,85,0A,8D,E2,02,8D,E4
5415 DATA 02,8D,E6,02,A9,01,8D,E5,02,A0,00,A5,05,85,00,91,0A,C8,A5,06
5420 DATA 85,01,91,0A,84,04,A5,04,C9,FF,F0,D2,A8,B1,0A,8D,E3,02,85,01
5425 DATA 88,B1,0A,8D,E1,02,85,00,88,84,04,20,C8,F1,A9,00,8D,E2,02,AD
5430 DATA E1,02,D0,DA,A5,00,85,02,A5,00,C9,F0,B0,24,A5,00,8D,E1,02,A5
5435 DATA 01,8D,E3,02,20,C8,F1,A9,00,8D,E2,02,AD,E1,02,D0,0D,A5,00,8D
5440 DATA E1,02,20,C8,F0,E6,00,4C,ED,9E,C6,00,A5,00,85,08,A5,02,85,00
5445 DATA C6,00,A5,00,C9,0C,90,1B,8D,E1,02,20,C8,F1,A9,00,8D,E2,02,AD
5450 DATA E1,02,D0,0B,A5,00,8D,E1,02,20,C8,F0,4C,21,9F,E6,00,A5,00,85
5455 DATA 09,A5,02,85,00,A5,01,F0,11,A5,00,85,02,A5,01,85,03,A5,09,85
5460 DATA 00,C6,01,20,7D,9F,A5,01,C9,C7,F0,11,A5,00,85,02,A5,01,85,03
5465 DATA A5,09,85,00,E6,01,20,7D,9F,4C,C3,9E,A5,00,C5,08,F0,0C,B0,5C
5470 DATA 90,08,A5,00,C5,08,F0,02,B0,52,8D,E1,02,A5,01,8D,E3,02,20,C8
5475 DATA F1,A9,00,8D,E2,02,AD,E1,02,F0,0D,E6,00,4C,87,9F,A5,00,C5,08
5480 DATA F0,02,B0,30,A4,04,C8,A5,00,91,0A,C8,A5,01,91,0A,84,04,A5,00
5485 DATA C5,08,F0,02,B0,1A,8D,E1,02,A5,01,8D,E3,02,20,C8,F1,A9,00,8D
5490 DATA E2,02,AD,E1,02,D0,C8,E6,00,4C,BF,9F,A5,02,85,00,A5,03,85,01
5495 DATA 60,EE
```



Vite fait Bien fait

CARACTÈRES GRAS

```
 1 FORI=1024TO1054STEP2:READA$:A=VAL("#"+A$):D
OKEI,A:NEXT:CALL1024:DATAB4A9
 2 DATA1185,A9,1085,E20,2004,40E,11E6,A0,10B1,
294A,111C,9110,8810,F4D0,5560
```

Jean FINKA

ADRESSES	MNÉMONIQUES	CODES	COMMENTAIRES
	initialisation lignes 1-30		
#9E82	LDA $Ø5	A5 Ø5	ILLEGAL QUANTITY ERROR SI X >= 24Ø OU X < 12 OU Y < Ø OU Y >= 2ØØ JSR $AØD2 SUR *ORIC-1*
	CMP $ #FØ	C9 FØ	
	BCS $ #1Ø	BØ 1Ø	
	CMP $ #ØC	C9 ØC	
	BCC $ #ØC	9Ø ØC	
	LDA $Ø6	A5 Ø6	
	CMP $ #ØØ	C9 ØØ	
	BCC $ #Ø6	9Ø Ø6	
	CMP $ #C8	C9 C8	
	BCS $ #Ø2	BØ Ø2	
	BCC $ #Ø4	9Ø Ø4	
	JSR $D336	2Ø 36 D3	
#9E9B	RTS	6Ø	ADRESSE DÉBUT DE PILE
	LDA $ #Ø4	A9 Ø4	
	STA $ØB	85 ØB	
	LDA $ #ØØ	A9 ØØ	
	STA $ØA	85 ØA	
	STA $Ø2E2	8D E2 Ø2	MISE A ZÉRO DE LA PARTIE HAUTE DE X, DE Y ET DE FD
	STA $Ø2E4	8D E4 Ø2	
	STA $Ø2E6	8D E6 Ø2	
	LDA $ # Ø1	A9 Ø1	FD <- - 1
	STA $Ø2E5	8D E5 Ø2	
	LDY $ # ØØ	AØ ØØ	X <- - XR X EMPILÉ
	LDA $Ø5	A5 Ø5	
	STA $ØØ	85 ØØ	
	STA ($ØA), Y	91 ØA	
	INY	C8	Y <- - YR Y EMPILÉ
	LDA $Ø6	A5 Ø6	
	STA $Ø1	85 Ø1	
	STA ($ØA), Y	91 ØA	
	STY $Ø4	84 Ø4	MISE A JOUR POINTEUR PILE
	ligne 4Ø		
#9EC3	LDA $Ø4	A5 Ø4	SI POINTEUR = −1 ALORS FIN
	CMP $ # FF	C9 FF	
	BEQ $ #D2	FØ D2	
	ligne 5Ø		
	TAY	A8	Y DÉPILÉ
	LDA ($ØA), Y	B1 ØA	
	STA $Ø2E3	8D E3 Ø2	
	STA $Ø1	85 Ø1	
	DEY	88	X DÉPILÉ
	LDA ($ØA), Y	B1 ØA	
	STA $Ø2E1	8D E1 Ø2	
	STA $ØØ	85 ØØ	MISE A JOUR POINTEUR PILE
	DEY	88	
	STY $Ø4	84 Ø4	
	ligne 6Ø		

ADRESSES	MNÉMONIQUES	CODES	COMMENTAIRES
	JSR $F1C8 LDA $#ØØ STA $Ø2E2 LDA $Ø2E1 BNE $#DA	20 C8 F1 A9 ØØ 8D E2 Ø2 AD E1 Ø2 DØ DA	JSR $F141 SUR *ORIC-1* SI POINT(X, Y) ALORS 4Ø
	ligne 7Ø à droite		
	LDA $ØØ STA $Ø2	A5 ØØ 85 Ø2	XS <– – X
	ligne 8Ø		
#9EED	LDA $ØØ CMP $#EF BCS $#24	A5 ØØ C9 EF BØ 24	SI X > 239 ALORS 11Ø
	LDA $ØØ STA $Ø2E1 LDA $Ø1 STA $Ø2E3	A5 ØØ 8D E1 Ø2 A5 Ø1 8D E3 Ø2	MISE EN PLACE PARAMÈTRES X ET Y DE FONCTION POINT
	ligne 9Ø		
	JSR $F1C8 LDA $#ØØ STA $Ø2E2 LDA $Ø2E1 BNE $#ØD	2Ø C8 F1 A9 ØØ 8D E2 Ø2 AD E1 Ø2 DØ ØD	JSR $F141 SUR *ORIC-1* SI POINT(X, Y) ALORS 11Ø
	ligne 1ØØ		
	LDA $ØØ STA $Ø2E1 JSR $FØC8	A5 ØØ 8D E1 Ø2 2Ø C8 FØ	**CURSET** X, Y, 1 JSR $FØ2D SUR *ORIC-1*
	INC $ØØ JMP $9EED	E6 ØØ 4C ED 9E	*X* <– – X+1 : ALLER EN 8Ø
	ligne 11Ø		
#9F17	DEC $ØØ LDA $ØØ STA $Ø8	C6 ØØ A5 ØØ 85 Ø8	XD <– – X–1
	LDA $Ø2 SRA $ØØ	A5 Ø2 85 ØØ	X <– – XS
	ligne 12Ø à gauche		
#9F21	DEC $ØØ	C6 ØØ	X <– – X–1
	ligne 13Ø		
	LDA $ØØ CMP $ØC BCC $#1B	A5 ØØ C9 ØC 9Ø 1B	SI X < 12 ALORS 16Ø
	ligne 14Ø		
	STA $Ø2E1 JSR $F1C8 LDA $#ØØ STA $Ø2E2	8D E1 Ø2 2Ø C8 F1 A9 ØØ 8D E2 Ø2	JSR $F141 SUR *ORIC-1* SI POINT (X, Y) ALORS 16Ø

ADRESSES	MNÉMONIQUES	CODES	COMMENTAIRES
	LDA $Ø2E1 BNE $ # ØB	AD E1 Ø2 DØ ØB	
	ligne 15Ø		
	LDA $ØØ STA $Ø2E1 JSR $FØC8	A5 ØØ 8D E1 Ø2 2Ø C8 FØ	**CURSET** X, Y, 1 JSR $FØ2D SUR *ORIC-1*
	JMP $9F21	4C 21 9F	ALLER EN 12Ø
	ligne 16Ø		
#9F44	INC $ØØ LDA $ØØ STA $Ø9	E6 ØØ A5 ØØ 85 Ø9	XG <- - X+1
	LDA $Ø2 STA $ØØ	A5 Ø2 85 ØØ	X <- - XS
	ligne 17Ø	voir dessus	
	LDA $Ø1 BEQ $ # 11	A5 Ø1 FØ 11	SI Y = Ø ALORS voir dessous
	ligne 18Ø		
	LDA $ØØ STA $Ø2	A5 ØØ 85 Ø2	XS <- - X
	LDA $Ø1 STA $Ø3	A5 Ø1 85 Ø3	YS <- - Y
	LDA $ØØ STA $ØØ	A5 Ø9 85 ØØ	X <- - XG
	DEC $Ø1 JSR $9F7D	C6 Ø1 2Ø 7D 9F	Y <- - Y-1 ss-prog TRONC COMMUN
	ligne 19Ø	voir dessous	
	LDA $Ø1 CMP $ # C7 BEQ $ # 11	A5 Ø1 C9 C7 FØ 11	SI Y = 199 ALORS 21Ø
	ligne 2ØØ		
	LDA $ØØ STA $Ø2	A5 ØØ 85 Ø2	XS <- - X
	LDA $Ø1 STA $Ø3	A5 Ø1 85 Ø3	YS <- - Y
	LDA $Ø9 STA $ØØ	A5 Ø9 85 ØØ	X <- - XG
	INC $Ø1 JSR $9F7D	E6 Ø1 2Ø 7D 9F	Y <- - YU+1 ss-prog TRONC COMMUN
	ligne 21Ø		
	JMP $9EC3	4C C3 9E	ALLER EN 4Ø
	ss-prog TRONC COMMUN		

ADRESSES	MNÉMONIQUES	CODES	COMMENTAIRES
	ligne 3ØØ		
# 9F7D	LDA $ØØ CMP $Ø8 BEQ $ # ØC BCS $ # 5C BCC $ # Ø8	A5 ØØ C5 Ø8 FØ ØC BØ 5C 9Ø Ø8	SI X > XD ALORS 39Ø SINON LABEL
	ligne 31Ø		
#9F87	LDA $ØØ CMP $Ø8 BEQ $ # Ø2 BCS $ # 52	A5 ØØ C5 Ø8 FØ Ø2 BØ 52	SI X > XD ALORS 39Ø
LABEL	STA $Ø2E1 LDA $Ø1 STA $Ø2E3	8D E1 Ø2 A5 Ø1 8D E3 Ø2	MISE EN PLACE PARAMÈTRES X ET Y DE FONCTION POINT
	ligne 32Ø		
	JSR $F1C8 LDA $ # ØØ STA $Ø2E2 LDA $Ø2E1 BEQ $ # ØD	2Ø C8 F1 A9 ØØ 8D E2 Ø2 AD E1 Ø2 FØ ØD	JSR $F141 SUR *ORIC-1* SI NON POINT (X, Y) ALORS 35Ø
	ligne 33Ø		
#9FA4	INC $ØØ JMP $9F87	E6 ØØ 4C 87 9F	X <– – X+1 ALLER EN 31Ø
	ligne 34Ø		
	LDA $ØØ CMP $Ø8 BEQ $ # Ø2 BCS $ # 3Ø	A5 ØØ C5 Ø8 FØ Ø2 BØ 3Ø	SI X > XD ALORS 39Ø
	ligne 35Ø		
# 9FB1	LDY $Ø4 INY	A4 Ø4 C8	GESTION POINTEUR PILE
	LDA $ØØ STA ($ØA), Y	A5 ØØ 91 ØA	X EMPILE
	INY LDA $Ø1 STA ($ØA), Y	C8 A5 Ø1 91 ØA	Y EMPILE
	STY $Ø4	84 Ø4	GESTION POINTEUR PILE
	ligne 36Ø		
#9FBF	LDA $ØØ CMP $Ø8 BEQ $ # Ø2 BCS $ # 1A	A5 ØØ C5 Ø8 FØ Ø2 BØ 1A	SI X > XD ALORS 39Ø
	LDA $Ø1 STA $Ø2E3	8D E1 Ø2 A5 Ø1 8D E3 Ø2	MISE EN PLACE PARAMÈTRES X ET Y DE FONCTION POINT
	ligne 37Ø		

	JSR $F1C8 LDA $ #ØØ STA $Ø2E2 LDA $Ø2E1 BNE $ #C8	2Ø C8 F1 A9 ØØ 8D E2 Ø2 AD E1 Ø2 DØ C8	JSR $F141 SUR *ORIC-1* SI POINT (X, Y) ALORS 33Ø
	ligne 38Ø		
	INC $ØØ JMP $9FBF	E6 ØØ 4C BF 9F	X <-- X+1 ALLER EN 36Ø
	ligne 39Ø		
#9FE1	LDA $Ø2 STA $ØØ	A5 Ø2 85 ØØ	X <-- XS
	LDA $Ø3 STA $Ø1	A5 Ø3 85 Ø1	Y <-- YS
	RTS	6Ø	FIN

Organisation de la page Ø

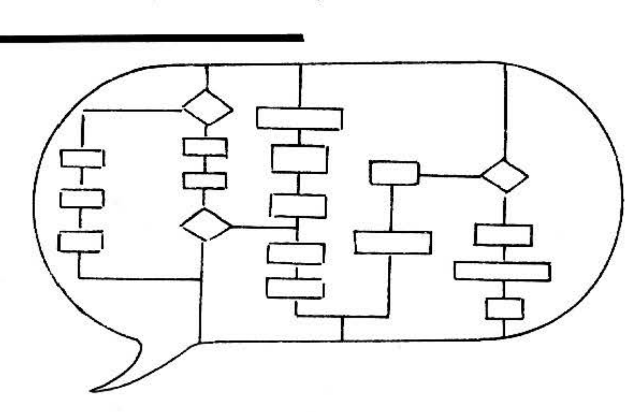
# ØØ	valeur de X
# Ø1	valeur de Y
# Ø2	valeur de XS
# Ø3	valeur de YX
# Ø4	pointeur
# Ø5	valeur de XR
# Ø6	valeur de YR
# Ø7	inutilisé
# Ø8	valeur de XD
# Ø9	valeur de XG
# ØA	adresse pile
# ØB	sur 16 bits

Organisation zone de passage des paramètres

# Ø2EØ	booléen de retour des fonctions
# Ø2E1	valeur du paramètre X
# Ø2E2	codée sur 16 bits
# Ø2E3	valeur du paramètre Y
# Ø2E4	codée sur 16 bits
# Ø2E5	valeur du paramètre FD
# Ø2E6	codée sur 16 bits

Adresses des fonctions utilisées

	ATMOS	*ORIC-1*
POINT	#F1C8	#F141
CURSET	#FØC8	#FØ2D
ILLÉGAL QUANTITY	#D336	#D2AØ





Programme Commenté

BULLETIN DE NOTES

par L. AUGUSTONI et P. CHICOURRAT

Plusieurs lecteurs se sont intéressés à ce sujet.

Vous êtes certainement nombreux à avoir mis au point un programme analogue. Pour ceux d'entre vous qui débutent nous vous proposons celui-ci. Les commentaires fournis peuvent vous permettre d'en tirer des idées de programmation. Tel quel, il est utilisable à l'école élémentaire pour dix mois et sept matières. La classe fictive inscrite comporte seulement 18 élèves. Il est proposé pour un ATMOS avec utilisation de cassette. Les adaptations à un lecteur de disquettes sont aisées. Nous lirons avec intérêt les remarques que ce sujet peut susciter.

```
 10 NE=18:NM=7:FORI=#BB80TO#BBA7:POKEI
,32:NEXT
 20 CLS:POKE618,10:DIMM$(7),N(NE,10,NM
+2),NOM$(NE):D$=CHR$(4):V$=CHR$(27)
 30 PAPER0:INK4:PRINT@10,11;D$V$"S"V$"
JBULLETIN DE NOTES "V$"P"D$
 40 GOSUB1730:GOSUB1920:GOTO790
 50 X=PEEK(#269):Y=PEEK(#268):PRINT@2,
21;"Calcul du total et de la moyenne"
 60 POKE#30E,81:FORII=1TONE:SOM=0:NN=0
 70 PLOT37,21,II-6*INT(II/6)+1
 80 FORKK=1TONM
 90 PRINT@38,21;M$(IM):IM=-(IM+1)*(IM<
>3)
 100 IFN(II,J,KK)>-1THENNN=NN+1:SO=SO+
N(II,J,KK)
 110 NEXTKK:N(II,J,NM+1)=SO
 120 IFNN>0THENX=SO/NN:GOSUB660:N(II,J
,NM+2)=X:GOTO140
 130 N(II,J,NM+2)=-1:N(II,J,NM+1)=-1
 140 NEXTII:POKE#30E,192:PRINT@2,21;CH
R$(14):PRINT@X,Y;:RETURN
 150 GOSUB160:PRINT:PRINT:RETURN
 160 GOSUB170:PRINTNO$;:RETURN
 170 NO=N(I,J,K):IFNO=-1THENNO$="   -":
RETURN
 180 NO$=CHR$(-32*(NO<100))+CHR$(-32*(
NO<10))
 190 NO$=NO$+CHR$(-48*(NO<1ANDNO>0))+M
ID$(STR$(NO),2)
 200 A$="":FORL=1TOLEN(NO$):K$=MID$(NO
$,L,1):IFK$="."THENK$=","
 210 A$=A$+K$:NEXTL:NO$=A$:RETURN
 220 GOSUB250:X=N(I,J,K):GOSUB660:N(I,
J,K)=X
 230 IFX>20THENEXPLODE:GOTO220
 240 GOTO150
 250 POKE618,3:T$="":X=PEEK(#269):Y=PE
EK(#268)-1:REPEAT
 260 GETR$:IFR$=CHR$(127)THENGOSUB370:
GOTO260
 270 IFR$="F"THENPULL:POP:POP:RETURN
 280 IFR$=","THENR$="."
 290 R=ASC(R$):IF(R<48ORR>57)ANDR<>45A
NDR<>13ANDR<>32ANDR<>46THENPING:GOTO26
0
 300 T$=T$+R$:PRINTR$;:POKE48,0
 310 IFLEN(T$)>6THENZAP:PRINT@X,Y;"
    "@X,Y;:T$="":GOTO260
 320 UNTILR=13ORR=32ORR=45:POKE618,10
 330 PRINT@X,Y;"          "@X,Y;:IFT$=CHR
$(13)THENZAP:GOTO250
 340 IFR=45THENN(I,J,K)=-1:RETURN
 350 IFR=32THENRETURN
 360 N(I,J,K)=VAL(T$):RETURN
 370 IFLEN(T$)>0THENPRINTR$;:T$=MID$(T
$,1,LEN(T$)-1)
 380 RETURN
 390 GETA$:R=-(A$=CHR$(10))+(A$=CHR$(1
1)):A=(A$=CHR$(8))-(A$=CHR$(9))
 400 I=I+R:J=J+A
 410 IFJ<0THENJ=9:I=I-1
 420 IFJ>9THENJ=0:I=I+1
 430 IFI<1THENI=NE
 440 IFI>NETHENI=1
 450 IF-ABS(A+R)THENRETURN
 460 POP:RETURN
 470 CLS:PAPER2:INK4:PRINT@7,4;"CHOIX
DU MOIS":PRINT:PRINT
 480 FORZ=0TO9:PRINTZ" - "MOI$(Z):NEXT
 490 PRINT@10,19;"QUEL MOIS?";:GETR$:J
=VAL(R$):IFJ<0ORJ>9THEN470
```

```
 500 PRINTJ:PRINT:PRINT"MOIS CHOISI: "V
$"S"MOI$(J)V$"R":WAIT50:RETURN
 510 CLS:PAPER6:INK4:PRINT@7,2;"CHOIX
DE L'ELEVE":PRINT
 520 FORZ=11TO10+NE:PRINTZ" - "NOM$(Z-
10):NEXTZ
 530 PRINT@10,23;"QUEL ELEVE?";:GETA$:
GETB$:A=VAL(A$):B=VAL(B$)
 540 A=10*A+B:IFA<11ORA>NE+10THEN510
 550 PRINTA:I=A-10:PRINT:PRINT"ELEVE C
HOISI;"V$"Q"NOM$(I)V$"V";:WAIT50:RETUR
N
 560 CLS:PAPER4:INK3:PRINT@7,2;"CHOIX
DE LA MATIERE":PRINT
 570 FORZ=1TONM:PRINT:PRINTZ" - "MAT$(
Z):NEXTZ
 580 PRINT@10,20;"QUELLE MATIERE?";
 590 GETR$:K=VAL(R$):IFK<1ORK>NMTHEN56
0
 600 PRINTK:PRINT:PRINT"MATIERE CHOISI
E :";
 610 PRINTV$"Q"MAT$(K)V$"T":WAIT50:RET
URN
 620 GOSUB640
 630 PRINTV$"D"V$"S"NOM$(I)V$"P":PRINT
:PRINT:RETURN
 640 CLS:PAPER0:INK2:IFJ=1ORJ=7THENPRI
NTV$"CMois d'";:ELSEPRINTV$"CMois de "
;
 650 PRINTMOI$(J):PRINT:RETURN
 660 IFX=-1THENRETURN ELSEX=INT(X*100+
.5)/100:RETURN
 670 SM=0:QM=0:FORK=1TONM:NN=0:SO=0
 680 FORJ=0TO9
 690 IFN(I,J,K)<>-1THENNN=NN+1:SO=SO+N
(I,J,K)
 700 NEXTJ:N(I,10,K)=-1
 710 IFNN>0THENX=SO/NN:GOSUB660:N(I,10
,K)=X:SM=SM+X:QM=QM+1
 720 NEXTK:N(I,10,NM+1)=-1:N(I,10,NM+2
)=-1
 730 IFQM>0THENN(I,10,NM+1)=SM:X=SM/QM
:GOSUB660:N(I,10,NM+2)=X
 740 RETURN
 750 X=1
 760 AD=#BB80+40*X:DOKE#278,AD+40:DOKE
#27A,AD:DOKE#27C,40*(27-X):POKE#27E,28
-X
 770 RETURN
 780 X=6:GOTO760
 790 GOSUB750:CLS:PAPER0:INK6
 800 PRINTD$@14,1;V$"JMENU"D$
 810 PRINT@6,5;"1 - CREER OU MODIFIER
UNE FICHE"
 820 PRINT@6,7;"2 - VISUALISER UNE FIC
HE"
 830 PRINT@6,9;"3 - VISUALISER PAR SEM
ESTRE"
 840 PRINT@6,11;"4 - SORTIE SUR IMPRIM
ANTE"
 850 PRINT@6,13;"5 - NOTES PAR MATIERE
S"
 860 PRINT@6,15;"6 - CORRECTION D'UNE
NOTE"
 870 PRINT@6,17;"7 - UTILISATION DE K7
"
 880 PRINT@6,19;"8 - FIN"
 890 PRINT@6,22;"VOTRE CHOIX?":GETR$:R
=VAL(R$):IFR<1ORR>8THEN790
 900 ONRGOSUB920,1020,1290,1580,1080,1
230,1960,2120
 910 GOTO790
 920 REM UNE FICHE
 930 CLS:GOSUB510:GOSUB470
 940 GOSUB620:FORK=1TONM:PRINTMAT$(K);
 950 PRINTSPC(16-LEN(MAT$(K)))V$"F:";:
GOSUB160:PRINTTAB(27)V$"C: ";
 960 GOSUB220
 970 NEXTK
 980 GOSUB50
 990 PRINT:PRINTTAB(21)V$"ATOTAL = ";:
K=8:GOSUB150
 1000 PRINT:PRINTTAB(19)V$"EMOYENNE =
";:K=9:GOSUB150
 1010 GOSUB390:GOTO940
 1020 CLS:GOSUB470:GOSUB510
 1030 GOSUB620:FORK=1TONM:PRINTMAT$(K)
SPC(15-LEN(MAT$(K)))": ";:GOSUB150:NEX
TK
 1040 PRINTSPC(16)"------"
 1050 PRINT:PRINTV$"A        TOTAL = "
;:GOSUB150
 1060 PRINTV$"E        MOYENNE = ";:K=9:
GOSUB150
 1070 GOSUB390:GOTO1030
 1080 REM NOTES PAR MATIERE
 1090 CLS:GOSUB470:GOSUB560
 1100 GOSUB640
 1110 PRINT:IFK=4ORK=7THENPRINTV$"GNot
es d'";:ELSEPRINTV$"GNotes de";
 1111 PRINTV$"A"MAT$(K):PRINT:PRINT
 1120 GOSUB780:NN=0:SO=0
 1130 FORI=1TONE
 1140 PRINTNOM$(I)TAB(20)V$"F:";:GOSUB
160:PRINTTAB(29)V$"C: ";:GOSUB220
 1150 IFX<>-1THENNN=NN+1:SO=SO+X
 1160 NEXTI
 1170 GETA$:CLS
 1180 PRINT@7,15;V$"EMOYENNE GENERALE
=";
 1190 IFNN=0THENPRINT" -":GOTO1210
 1200 X=SO/NN:GOSUB660:N(0,0,0)=X:I=0:
JJ=J:J=0:K=0:GOSUB150:J=JJ
 1210 GOSUB750
 1220 GOTO50
 1230 REM CORRECTION D'UNE NOTE
 1240 GOSUB510:GOSUB470:GOSUB560:GOSUB
620
 1250 INK6:PRINT"  "MAT$(K)SPC(15-LEN(
MAT$(K)))": ";:GOSUB150:PRINT
 1260 PRINTV$"E NOUVELLE NOTE  : ";:GO
SUB220
 1270 GOTO50
 1290 REM VISUALISER PAR SEMESTRE
 1300 CLS:PRINT@2,7;"VOULEZ-VOUS UN EL
EVE PARTICULIER?"
 1310 PRINT@12,15;"OUI ou NON?";
 1320 GETR$:PRINTR$:IFR$="O"ORR$="o"TH
ENGOSUB510:DR=-1:GOTO1340
 1330 FORI=1TONE
 1340 FORS=0TO1
 1350 CLS:PAPER3:INK4
 1360 PRINT:PRINTV$"A"NOM$(I)
 1370 PRINT@12,3;D$V$"J"M$(S+4)" SEMES
TRE: "D$
 1380 PRINT:PRINT:PRINTM$(S+6):PRINT
```

```
 1390 FORK=1TONM+2
 1400 IFK=8THENPRINTV$"AT";:GOTO1430
 1410 IFK=9THENPRINTV$"Em";:GOTO1430
 1420 PRINT" "LEFT$(MAT$(K),1);
 1430 FORJ=0TO4
 1440 PRINTTAB(5+7*J);:J=J+5*S:GOSUB16
0:J=J-5*S
 1450 NEXTJ:PRINT:PRINT:NEXTK:GETA$
 1460 IFA$="F"THENS=1:NEXTS:I=NE:GOTO1
570
 1470 NEXTS
 1480 CLS:PRINT:PRINTV$"A"NOM$(I):PRIN
T@12,3;D$V$"JMOYENNES ANNUELLES "D$
 1490 GOSUB670:J=10
 1500 PRINT@2,7;:FORK=1TONM
 1510 PRINTMAT$(K)SPC(16-LEN(MAT$(K)))
": ";:GOSUB150:NEXTK
 1520 PRINTSPC(18)"------"
 1530 PRINTV$"A           TOTAL = ";:GOS
UB150
 1540 PRINTV$"E         MOYENNE = ";:K=N
M+2:GOSUB150
 1550 GETA$
 1560 IFDRTHENDR=0:RETURN
 1570 NEXTI:RETURN
 1580 REM EDITION SUR IMPRIMANTE
 1590 GOSUB510:CLS:PAPER0:INK5:GOSUB63
0
 1600 PRINT:PRINT"SORTIE SUR IMPRIMANT
E DU TABLEAU"
 1610 PRINT:PRINT"RECAPITULATIF ANNUEL
":GOSUB670
 1620 LPRINTCHR$(14)NOM$(I)CHR$(15)TAB
(30)CHR$(14)"NOTES DE L'ANNEE"CHR$(15)
 1630 LPRINT:LPRINT
 1640 LPRINTM$(6)M$(7)"     MOY":LPRINT
 1650 FORK=1TO9
 1660 IFK=8THENLPRINT:LPRINT"T";:GOTO1
690
 1670 IFK=9THENLPRINT"m";:GOTO1690
 1680 LPRINTLEFT$(MAT$(K),1);
 1690 FORJ=0TO10
 1700 LPRINTTAB(3+7*J);:GOSUB170:LPRIN
TNO$;
 1710 NEXTJ:LPRINT:LPRINT
 1720 NEXTK:LPRINT:LPRINT:LPRINT:RETUR
N
 1730 REM LECTURE DES DONNEES
 1740 M$(0)="-":M$(1)="\":M$(2)="|":M$
(3)="/":M$(4)="PREMIER":M$(5)="SECOND"
 1750 M$(6)="     SEP    OCT    NOV
 DEC    JAN"
 1760 M$(7)="     FEV    MAR    AVR
 MAI    JUN"
 1770 FORZ=0TO9:READMOI$(Z):NEXTZ
 1780 FORZ=1TONM:READMAT$(Z):NEXTZ
 1790 FORZ=1TONE:READNOM$(Z):NEXTZ
 1800 J=#B570:FORI=#B560TO#B568:POKEJ,
PEEK(I):J=J+1:NEXT
 1810 RETURN
 1830 DATAseptembre,octobre,novembre,d
ecembre,janvier,fevrier
 1840 DATAmars,avril,mai,juin
 1850 DATASOIN,LECTURE,GRAMMAIRE,ORTHO
GRAPHE,CONJUGAISON,MATHEMATIQUES,EVEIL
 1860 REM PAS PLUS DE 18 CARACTERES PO
UR NOM ET PRENOM
 1870 DATAASTRUC Pierre,BERTHIER Chris
tian,CHOURTEL Jacques,FRESBEAU Yves
 1880 DATAGRAJON Luce,HERTOL Frederic,
JOULE Genevieve,KRONYCK Paule
 1890 DATALEBON Louis,MARCHAL Henri,PR
OTAIN Julie,PUVION Colette
 1900 DATARATEAU Gerald,RINCENT Lydie,
ROUANEY Denis,RUTON Beatrice
 1910 DATASARTONI Lucette,TRESOR Olivi
er
 1920 FORI=0TONE:FORJ=0TO10:FORK=0TONM
+2
 1930 N(I,J,K)=-1
 1940 NEXT:NEXT:NEXT
 1950 RETURN
 1960 REM UTILISATION DE K7
 1970 CLS:INK5:PRINT@3,7;"1 - RECUPERE
R LES NOTES"
 1980 PRINT@3,11;"2 - SAUVEGARDER LES
NOTES"
 1990 PRINT@6,17;"VOTRE CHOIX?":
 2000 GETR$:PRINTR$:R=VAL(R$):IFR<1ORR
>2THEN1960
 2010 ONRGOTO2020,2070
 2020 CLS:PRINT@2,10;V$"CMETTRE LE K7
EN LECTURE"
 2030 PRINT@2,12;V$"CAU BON ENDROIT"
 2040 PRINT@2,1V$"CPRESSER UNE TOUCHE
DU CLAVIER":GETR$
 2050 RECALL N,"NOTES"
 2060 RETURN
 2070 CLS:PRINT@2,10;V$"AMETTRE LE K7
SUR ENREGISTREMENT"
 2080 PRINT@2,12;V$"ARELEVER LE COMPTE
UR"
 2090 PRINT@2,14;V$"APRESSER UNE TOUCH
E DU CLAVIER":GETR$
 2100 STORE N,"NOTES"
 2110 RETURN
 2120 CALL#F8D0:CLS:END
```

10-30 - Initialisation :

10 NE, nombre d'élèves. NM, nombre de matières.

20 M$(7) voir en 1740.
N (NE, 10, NM + 2) tableau de stockage des notes, des moyennes, des totaux.
N (I, J, K) I indice élève, J indice mois, K indice matière.
NOM$(NE) tableau des noms et prénoms.

30 Affichage du titre. Signalons ici que l'usage abondant de la couleur prévu risque de géner ceux qui utilisent un écran monochrome. Il leur faudra effectuer les modifications indispensables.

40 Envoi en 1730 pour la lecture des données, la mise des variables dans les divers tableaux.
La ligne 1800 modifie le dessin du point : à l'écran il aura l'aspect d'une virgule, cependant pour les calculs, l'*ORIC* continue de considérer ce caractère comme un point. La ligne 200 est rendue nécessaire par ce choix.

183Ø-191Ø **DATA**s, on veillera en écrivant les noms et prénoms à ne pas dépasser 18 caractères, sinon l'affichage serait perturbé.

192Ø-195Ø. Ce module ne doit être utilisé qu'une fois en début d'année. Il met le nombre (– 1) dans toutes les variables N (I, J, K). A l'écran, ou sur l'imprimante un tiret apparaît pour signaler l'absence de note. La valeur (– 1) a été choisie pour mémoriser l'absence de note. Branchement en 79Ø à l'affichage du MENU.

5Ø-14Ø - S/P - Calcul des totaux et des moyennes

Puisque ce calcul se fait rapidement, il est refait systématiquement à chaque modification.

5Ø On mémorise les coordonnées actuelles du curseur dans les variables X et Y.

6Ø **POKE** 30E, 81 inhibe le clavier, l'ordinateur calcule plus vite. Noter II indice élève provisoire. NN, nombre de notes **SOM ou SO,** total.

7Ø On met en (37,21) un attribut de couleur variable indexé sur la variable II.

8Ø KK, indice matière provisoire.

9Ø On affiche l'un des 4 signes — \ | / successivement et on recommence, donnant ainsi l'impression d'un petit moulin qui tourne.

La formule IM = – (IM + 1) * (IM < > 3) mérite explications :

En ligne 174Ø vous voyez à quoi correspond M$(Ø) à M$(3).

si IM = **Ø**

alors IM = – (**Ø** + 1) * (**Ø** < > 3)
comme (**Ø** < > 3) est vraie, cette expression vaut (– 1)
d'où IM = (– 1) * (– 1)
IM = 1
on affiche alors M$(1) c'est-à-dire : \

si IM = **1**

alors IM = – (**1** + 1) * (**1** < > 3)
comme (**1** < > 3) est vraie, cette expression vaut (– 1)
d'où IM = (– 2) * (– 1)
IM = 2
on affiche alors M$(2) c'est-à-dire : |

si IM = **2**

alors IM = – (**2** + 1) * (**2** < > 3)
comme (**2** < > 3) est vraie, cette expression vaut (– 1)
d'où IM = (– 3) * (– 1)
IM = 3
on affiche alors M$(3) c'est-à-dire : /

si IM = **3**

alors IM = – (**3** + 1) * (**3** < > 3)
comme (**3** < > 3) est fausse, cette expression vaut Ø
d'où IM = (– 4) * Ø
IM = Ø
on affiche alors M$(Ø) c'est-à-dire : —

Ce calcul booléen permet une programmation plus compacte qu'avec des **IF... THEN** et d'exécution plus rapide. La programmation en langage machine utilise souvent un tel procédé.

1ØØ En l'absence de note on va directement en 11Ø.
On compte le nombre de notes (NN) et leur somme (SO).

11Ø La somme est placée dans N (II, J, NM + 1) en fin de boucle.

12Ø La moyenne (si elle existe) est calculée, formatée (GOSUB 66Ø) et affectée à N (IJ, J, NM + 2). X étant une variable provisoire.

13Ø Au cas où aucune note n'existe, la moyenne prend la valeur (– 1).

14Ø Quand on a calculé la moyenne pour tous les élèves et pour toutes les matières, et cela, pour un mois donné, on réactive le clavier par **POKE** # 30E, 192. On efface la ligne par **CTRL** N = **CHR**$(14). On place le curseur en (X,Y), coordonnées mémorisées en ligne 5Ø.

15Ø-21Ø - Sous-programme d'affichage d'une note

Entrée en 15Ø, on saute 2 lignes avant le retour.

Entrée en 16Ø, on affiche à l'écran la note (NO$) et on s'en retourne.

17Ø Si la note est (– 1), NO$ contient un tiret et c'est fini, on revient.

Sinon en 18Ø-19Ø on formatte le nombre pour un affichage en colonne. Pour comprendre ces deux lignes il faut savoir que **CHR**$(Ø) ne fait rien.

Si (NO < 1ØØ) est vraie alors l'expression vaut (– 1) et (– 32) * (– 1) = 32 et **CHR**$(32) est un espace.

De même avec (NO < 1Ø).

Quand NO < 1Ø on sort de la ligne 18Ø avec une chaîne NO$ qui contient 2 espaces.

Si NO est compris entre ∅ et 1, *ORIC* ne met pas de zéro devant le "point de décimalisation" l'expression (− 48 *(NO < 1 AND NO > ∅)) prend la valeur (− 48) * (− 1) = 48 dans ce cas et **CHR**$(48) est le chiffre ∅. On oblige l'écriture d'un zéro.

Il est temps d'afficher les chiffres significatifs.

MID$** (**STR$ (NO),2) nous débarasse du blanc généré automatiquement par l'*ORIC* devant les nombres positifs... Ouf !

Encore une coquetterie, en 2∅∅-21∅ on remplace le point, s'il s'en trouve un dans la chaîne NO$, par une virgule. (Attention ! Si vous listez après un **RUN** vous ne verrez pas la différence entre point et virgule ligne 2∅∅, sauf après avoir fait **CALL** # F8D∅).

Cette préparation de NO$ est utilisée pour l'écran ou l'imprimante.

22∅-38∅ - Sous-programme de saisie des caractères frappés au clavier.

22∅-24∅ aiguillages et test interdisant les notes supérieures à 2∅.

25∅ **POKE** 618,3 pour rétablir le curseur sonore et visible.
X et Y sont utilisées comme précédemment.
Boucle **REPEAT**... **UNTIL** (25∅-32∅).

Elle sert à donner l'impression d'un **INPUT** alors qu'on utilise **GET.** On construit T$.

Comme vous voyez en ligne 32∅, on n'en sort que par appui sur **RETURN** ou **ESPACE** ou − on rend alors le curseur à nouveau invisible et le clavier muet par **POKE** 618,1∅.

Que se passe-t-il dans la boucle ?

En 26∅ : lecture du caractère, affectation à R$.

Si c'est **DEL** on efface le caractère précédemment enregistré. Ceci se fait en 37∅ :

On écrit le caractère R$, comme c'est **DEL,** le curseur recule d'un cran et efface le caractère précédent. On change le contenu de T$, en supprimant le dernier caractère de cette chaîne.

Si l'on tape "F", alors là, c'est de l'acrobatie !

Les puristes supprimeront cette ligne horrifiante...

Un **PULL** et 2 fois **POP** ! Quelle sortie brutale !

C'est pratique lorsqu'on a demandé l'exécution d'un affichage bien long et que l'on constate qu'on s'est trompé. Aux essais... cela marche !

Alors, à vous de choisir : en gardant la ligne 27∅, F (comme **FIN**) vous renverra au **MENU.**

28∅ Si vous tapez un nombre avec une virgule, on prévient *ORIC* qu'il s'agit d'un point, mais vous verrez toujours une virgule à l'écran.

29∅ On attend certains caractères... pas les autres sinon... PING !

3∅∅ On accumule les signes, on les affiches (**POKE** 48,∅ est là pour contrôler la position du curseur, faites-nous confiance).

31∅ Vous avez droit à 2 chiffres avant la virgule, et 2 chiffres après. On vous offre 6 caractères par pure faiblesse : 5 cela devrait suffire. Au-delà... ZAP ! et on efface votre vilain travail, on oublie les signes accumulés dans T$.

32∅ Fin de saisie d'une note. Effacement du curseur.

33∅ On efface et on se prépare à écrire la note comme il convient. On proteste si vous avez entré la chaîne vide !

34∅ Si vous avez signalé par un tiret l'absence de note, on mémorise (− 1). Retour.

35∅ Si vous avez appuyé sur la barre d'espace on considère que vous n'entrez pas de nouvelle note. Retour.

36∅ Autrement la nouvelle valeur remplace l'ancienne et retour.

39∅-46∅ - Sous-programme de contrôle des indices I et J, c'est-à-dire élève et mois par les touches fléchées.

39∅ Saisie d'un caractère, affectation à A$.

- Si c'est ↑ , de code 11,
  alors R = − ∅ + (− 1) = − 1 et A = ∅
  en ligne 4∅∅ l'indice élève est décrémenté, l'indice du mois ne change pas.
- Si c'est ↓ de code 1∅,
  alors R = − (− 1) + ∅ = 1 et A = ∅.
  L'indice élève est incrémenté, l'indice du mois ne change pas.
- Si c'est → de code 9,
  alors R = ∅ et A = ∅ − (− 1) = 1
  on ne change pas d'élève, on passe au mois suivant.
- Si c'est ← de code 8,
  alors R = ∅ et A = − 1 − ∅ = (− 1)
  on ne change pas d'élève, on revient au mois précédent.

41∅ Si l'on est en septembre et qu'on appuie sur ← alors on va en juin et on passe à l'élève précédent.

42∅ Si l'on est en juin et qu'on appuie sur → alors on va en septembre et on passe à l'élève suivant.

43∅ Si l'on est en début de liste et qu'on demande l'élève précédent, on va en fin de liste.

440 Si l'on est en fin de liste et qu'on demande l'élève suivant, on va au début de la liste.
450 Si A = − 1 ou R = − 1, ou A = − 1 ou R = 1, alors − ABS (A + R) = (− 1), le test est satisfait on revient où l'on était, traiter le cas d'un autre élève ou d'un autre mois.
460 Sinon, on saute un retour et l'on revient au **MENU.**

Nous espérons vous donner envie d'employer des méthodes analogues dans vos propres programmes : c'est court et efficace.

470-500 - S/P choix du mois.

510-550 - S/P choix de l'élève.

560-610 - S/P choix de la matière.

620-650 - S/P affichage du mois et/ou du nom.

Si l'on entre en 620 on affiche le mois puis le nom, si l'on entre en 630 seulement le nom, en 640 seulement le mois.

660- - S/P d'arrondi automatique.

670-740 -Sous-programme de calcul du total et de la moyenne sur dix mois.

SM = somme en une matière.
QM = nombre de matières.
NN = nombre de notes.
SO = total.

750-910 - MENU et AIGUILLAGE PRINCIPAL

750-780 Création d'une fenêtre X = 6, on fige les 6 lignes supérieures. X = 1, on restitue tout l'écran.
790 Envoi à la sous routine restituant l'écran. Le reste est très facile à comprendre.

920-1010 - Création ou modification d'une fiche

On appelle fiche les notes d'un élève, un mois donné.

930 Choix de l'élève puis du mois.
940 Affichage du mois, puis du nom, enfin des matières et de l'ancienne note (ou d'un tiret). Envoi au module de saisie à chaque fois.
970 Inscription de la nouvelles note s'il y a lieu.
980 Mise à jour des totaux et de la moyenne.
990-1000 Affichage des totaux et de la moyenne.
1010 S/P en 390 glissement d'un mois à l'autre, d'un élève à l'autre à l'aide des flèches. Si l'une des 4 flèches a été actionnée retour en 940 sinon voir ligne 460, esquisse d'un retour et renvoi au **MENU.**

1020-1070 - Visualisation d'une fiche

Comme le module précédent, mais affichage seul.

C'est rapide. L'emploi des flèches pour se déplacer dans le fichier est particulièrement agréable ici.

1080-1220 - Enregistrement ou modification de notes

Un mois donné pour une matière.

1090 Choix du mois et de la matière.
1100-1111 Affichage de l'en tête.
1120 On fige les 6 lignes du haut. NN (nombre de notes), SO (total).
1130-1160 Pour chaque élève, affichage du nom, de l'ancienne note (ou du tiret). Saisie de la nouvelle et affichage. Calcul du total et du nombre de notes.

Rappel : **F** pour arrêter en cours de route.
espace pour conserver l'ancienne note.
− pour indiquer l'absence de note.
nouvelle note suivie de **RETURN.**

1170-1200 Après le dernier élève, l'appui sur n'importe quelle touche (**GET** A$ sans test sur A$) provoque l'effacement de l'écran et l'affichage de la moyenne de la classe pour le mois dans la matière.
1210 On restitue tout l'écran.
1220 On passe la main au module de calcul des moyennes qui renverra au menu. (Attention ! L'emploi de F renvoie au menu, les moyennes et les totaux sont faussés. Revenir pour inscrire les notes).

1230-1270 - Correction d'une note

Compte-tenu des explications précédentes, ce court module est facile à comprendre.

1290-1570 - Bulletin de notes complet en 3 écrans

1 - Premier semestre.
2 - Second semestre.
3 - Moyennes générales.

1330-1570 Boucle élève.
1340-1470 Boucle semestre.
1300-1320 Choix éventuel d'un élève précis. DR sert de drapeau qui sera utilisé en 1560.
1430-1450 Pour cinq mois à la fois, affichage des notes avec tabulations à respecter.
1460 Sortie du module par appui sur F. (Remarquez comment on respecte les boucles).
1480-1540 Affichage des moyennes générales annuelles.
Remarquez que J = 10 correspondrait à

juillet, mais est utilisé pour la moyenne annuelle par matières.

1550 Appuyer sur une touche pour continuer.

1570 Fin de module.

1580-1720 - Édition sur imprimante

1620 **CHR**$(14) et **CHR**$(15) sont des codes qui peuvent varier d'une imprimante à l'autre. **CHR**$(14) passage en double largeur. **CHR**$(15) retour à la largeur usuelle. On peut les supprimer.

1640 M$(6) est en 1750, M$(7) en 1760.

1700 La tabulation est calculée sur 80 colonnes.

1960-2110 - Utilisation du magnétophone

Facile à comprendre.

En cas d'utilisation de disquette il est préférable d'utiliser deux lettres au lieu d'une NT (I,J,K) et **! STORE** NT, **"NOTES. DAT"**.

Vous pouvez aussi utiliser un fichier à accès séquentiel. Cela ne pose pas de problème pour les possesseurs de microdisques.

2120 - Fin

On reconfigure les caractères.



DELETE

par Dominique CHARMEAU

Notice explicative de DELETE

Ce programme *BASIC* compatible *ORIC-1* et *ATMOS* permet la suppression rapide de la portion de programme comprise entre deux numéros de ligne. Ce programme est très utile par exemple lorsqu'on a un programme très long et qu'on décide de le transformer en plusieurs programmes indépendants. En effet, pour supprimer les lignes entre 5000 et 7000, il est fastidieux d'écrire 5000 puis **RETURN,** 5010 **RETURN,** 5020... 6990 **RETURN.** De plus par cette méthode, on ne supprime que les lignes multiples de 10.

Avec ce programme, il suffira de donner le numéro de la ligne de départ (5000), celui de la dernière ligne à supprimer (6999) puis de taper 5000 suivi de **RETURN.**

```
63980 INPUT"DELETE DE";D:INPUT"A";F
63982 A=0:DOKE0,1281
63984 REPEAT:A=DEEK(A):UNTIL DEEK(A+2)=D
63986 E=DEEK(A+2):IF E>F THEN END
63988 PRINT"TAPER:"E"PUIS RETURN"
63990 B=DEEK(A):IFDEEK(B+2)>F THEN END
63992 REPEAT:C=DEEK(B):POKE B-1,32
63994 DOKEB,8250:DOKEB+2,8250
63996 B=C:UNTIL DEEK(C+2)>F
63998 DOKE A,C
```

Ce programme doit être chargé à la suite du programme à modifier (on peut le faire par la fonction merge (cf *MICR'ORIC* pour *ORIC-1,* ou bien le taper car il est très court). Puis il faut faire **RUN** 63980.

Explication : Ce programme recherche l'adresse de la première ligne E (E ⩾ D) à supprimer puis transforme les lignes suivantes jusqu'à la ligne F : il remplace les O, adresse ligne suivante, numéro de ligne par espace, espace : , espace : (cf. articles chirurgie en RAM).

Toutes les lignes de E jusqu'à F sont ainsi mises bout à bout pour ne former qu'une seule (et longue !) ligne E. Il suffit alors de supprimer cette ligne E.





LES VARIABLES * *

par Jacques-François WADEL

Vous avez pu lire dans *MICR'ORIC* n° 5 l'article de M. Pierre LEDAIN, la suite de cet article est dans le n° 7, puis MM. Guy JUY et Christian MAGRIN ont décortiqué le codage des nombres à virgules flottante, le sujet est loin d'être épuisé. L'*ORIC* gère 4 types de variables : à virgule flottante, entière, chaîne de caractères et celles définies par **DEF FN** sur lesquelles il faudra revenir.

L'exploration de la zone mémoire est possible mais lente et malaisée, le programme que nous vous proposons ici vous rendra de grands services pour la correction, la mise au point de vos programmes. Mis en place par exemple entre 60000 et 60090 accolé à votre programme, un **GOTO** 60000 vous donnera à l'écran ou sur votre imprimante au choix la liste des variables utilisées et leur contenu, n'est-ce pas merveilleux ? Dans l'exemple proposé nous avons écrit lignes 10 à 60 des variables quelconques, voyez comment le programme les détecte et les affiche.

```
10 A=1234:B=4321
20 A$="ABCDEFGHIJ"
30 BS$="BSBSBSBSBSBSBS"
35 DEFFNAA(X)=A+A+A+A+A+A
40 XX%=1234
50 AB=123456:BA=654321
60 E5=555:E%=421:F$=STR$(AB)
60000 Z0=1281:IFZZ=1THENPOKE#2F1,128
60005 Z1=DEEK(Z0):IFZ0>Z1THENSTOP
60010 IFDEEK(Z0+2)=60075THEN60020
60015 Z0=Z1:GOTO60005
60020 FORZ1=DEEK(#9C)TODEEK(#9E)-1STEP7
60025 IFPEEK(Z1)>127THEN60036
60030 IFPEEK(Z1+1)>127THENZ0$="CHAINE    ":Z3=0:Z4=128:GOTO60040
60035 Z3=0:Z4=0:Z0$="VIRG FLOT":GOTO60040
60036 IFPEEK(Z1+1)>127THENZ0$="ENTIER    ":Z3=128:Z4=128:GOTO60040
60037 Z3=128:Z4=0:Z0$="* DEF FN "
60040 Z2=Z0+5:Z1$=""
60045 POKEZ2,PEEK(Z1)-Z3:Z1$=CHR$(PEEK(Z1)-Z3)
60050 IFPEEK(Z1+1)=Z4THEN60060
60055 Z2=Z2+1:POKEZ2,PEEK(Z1+1)-Z4:Z1$=Z1$+CHR$(PEEK(Z1+1)-Z4)
60056 IFZ3=128ANDZ4=0THEN60085
60060 IFZ4=0THEN60065
60061 Z2=Z2+1:Z5=37:IFZ3=0THENZ5=Z5-1
60062 Z1$=Z1$+CHR$(Z5):POKEZ2,Z5
60065 POKEZ2+1,58:POKEZ2+2,157
60070 PRINTZ0$"--"Z1$"==";
```

```
60075 PRINTZZ:REMREM*******************
60080 NEXTZ1:END
60085 PRINTZ0$"--"Z1$"--voir listing--"
60090 NEXTZ1
```

Pour obtenir la sortie sur imprimante il suffit de mettre la variable **ZZ** à **1.** Elle sert de drapeau en 60000.

Par exemple, après avoir fait tourner votre programme, vous l'interrompez par **CTRL C,** vous écrivez ZZ = 1, vous faites **GOTO** 60000 et la liste des variables dans leur état actuel sort sur l'imprimante (voir ci-contre).

Ce programme doit sa faible longueur à la ligne 60075, elle évolue en cours d'emploi : vérifiez en listant plusieurs fois. Au début, veillez à ce que cette ligne comporte une instruction **PRINT** puis n'importe quoi occupant 20 octets au minimum, c'est essentiel. Les lignes 60040 à 60065 ont pour rôle d'inscrire dans cette zone ce qu'il faut quand il faut... Les lignes 60000 à 60015 se chargent de trouver l'adresse de ce **PRINT.** La ligne 60020 explore la mémoire de #9C à #9E avec un pas de 7 (**STEP 7**).

Chaque variable ainsi trouvée est sélectionnée selon son type (entière, virgule flottante, chaîne, **DEF FN**) et est affectée à Z0S, Z1$, et le **PRINT** est modifié par **POKE.**

```
VIRG FLOT--ZZ== 1
VIRG FLOT--A== 1234
VIRG FLOT--B== 4321
CHAINE   --A$==ABCDEFGHIJ
CHAINE   --BS$==BSBSBSBSBSBSBS
* DEF FN --AA--voir listing--
VIRG FLOT--X== 0
ENTIER   --XX%== 1234
VIRG FLOT--AB== 123456
VIRG FLOT--BA== 654321
VIRG FLOT--E5== 555
ENTIER   --E%== 421
CHAINE   --F$== 123456
VIRG FLOT--Z0== 2010
VIRG FLOT--Z1== 2192
VIRG FLOT--Z3== 0
VIRG FLOT--Z4== 0
CHAINE   --Z0$==CHAINE
VIRG FLOT--Z2== 2016
CHAINE   --Z1$==Z1$
VIRG FLOT--Z5== 36
```

PROMOTION COMPOSANTS ÉLECTRONIQUES

BON DE COMMANDE

(quantité minimum 1 paquet)

Quantité par paquet	CIRCUITS INTÉGRÉS	T.T.C.	Quantité	Total
25	SN 74LS00	100		
25	SN 74LS02	100		
25	SN 74LS04	120		
25	SN 74LS08	120		
25	SN 74LS10	100		
25	SN 74LS14	150		
25	SN 74LS20	100		
25	SN 74LS30	100		
25	SN 74LS42	140		
25	SN 74LS74	140		
25	SN 74LS161	175		
25	SN 74LS163	175		
25	SN 74LS164	175		
25	SN 74LS175	175		
20	SN 74LS240	200		
20	SN 74LS241	200		
20	SN 74LS244	200		
20	SN 74LS245	250		
25	SN 74LS367	250		
25	SN 74LS368	250		
20	SN 74LS373	300		
20	SN 74LS374	300		
25	ULN 2003A	250		
25	ULN 2004A	250		
25	ULN 2804A	400		
25	UDN 2981A	400		
10	UDN 6118A	200		
10	UDN 6128A	200		
	TRANSISTORS			
10	TIP 29	45		
10	TIP 30	45		
10	TIP 31	45		
10	TIP 32	45		
10	TIP 41	65		
10	TIP 42	65		
10	TIP 50	90		

Quantité par paquet	TRANSISTORS (suite)	T.T.C.	Quantité	Total
10	TIP 53	90		
10	TIP 110	90		
10	TIP 115	90		
10	TIP 117	90		
10	TIP 136	90		
10	TIP 147	90		
10	TIP 2955	90		
10	2N 2219	35		
10	2N 2222A	35		
10	2N 2907A	35		
10	2N 2905	35		
10	2N 2906	35		
10	2N 3055	80		
	MÉMOIRES MICROPROCESSEURS			
8	4116/15	140		
8	4164/15	300		
2	2708	120		
2	2516	100		
2	2732	150		
2	2764	190		
2	6502	190		
2	Z80	190		
2	27128	240		
	SPÉCIAL			
2 500	Résistances 1/4 W 25 valeurs de 100 pièces/valeur	300		
2 500	Résistances 1/2 W 25 valeurs de 100 pièces/valeur	300		
250	Résistances bobinés 3 W 10 valeurs de 25 pièces/valeur	600		
250	Résistances bobinés 7 W 10 valeurs de 25 pièces/valeur	700		

Quantité par paquet	SPÉCIAL (suite)	T.T.C.	Quantité	Total
100	Condensateurs chimiques 10 valeurs de 10 pièces/valeur	200		
100	1N 4002	50		
100	1N 4004	60		
100	1N 4007	70		
100	1N 4148	30		
10	Ponts 1 A - 200 V	40		
10	Ponts 10 A - 200 V	150		
10	Porte fusible 5× 20 face avant Réf. F296	60		
10	Prises européennes 2 + T femelle Réf. P580	60		
10	Prises européennes 2 + T Mâle Réf. P587	90		
10	Régulateurs TO 220 5 V	75		
1	prise filtrée 3A	78		
10	Supports CI 24 broches	30		
10	Supports CI 40 broches	50		
1	Soudure 500 g	100		
	SPÉCIAL ORIC-1/ATMOS			
1	Clavier ATMOS BNO 138	300		
1	ROM V1.1 IC 0072	220		
1	ULA 10017 IC 0069	220		
1	TBP 24S10 IC 0071	20		
1	CMOS 4051 IC 0059	6		
1	AY 8912 IC 0067	69		
1	µP 6502 IC 0066..........	75		
1	PIA 6522 IC 0049	85		

TOTAL 1 ☐ **TOTAL 2** ☐ **TOTAL 3** ☐

Report **TOTAL 1** ☐

TOTAL 2 ☐

Port + **25 F**

TOTAL A PAYER ☐

Mon adresse : ____________________

____

____

Ci-joint mon règlement par chèque.

Signature :

(des PARENTS pour les Mineurs)

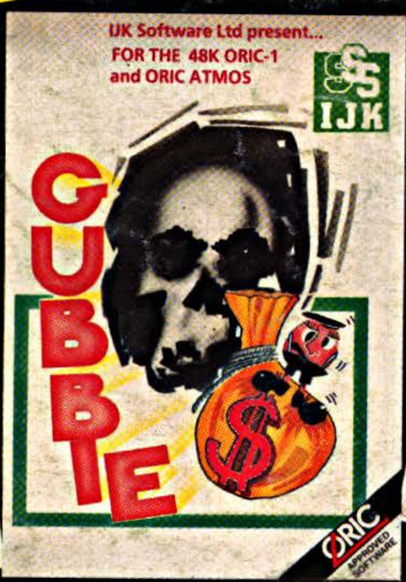
POUR TOUT RENSEIGNEMENT SUR LES AUTRES COMPOSANTS DISPONIBLES TÉLÉPHONER AU 599.22.22

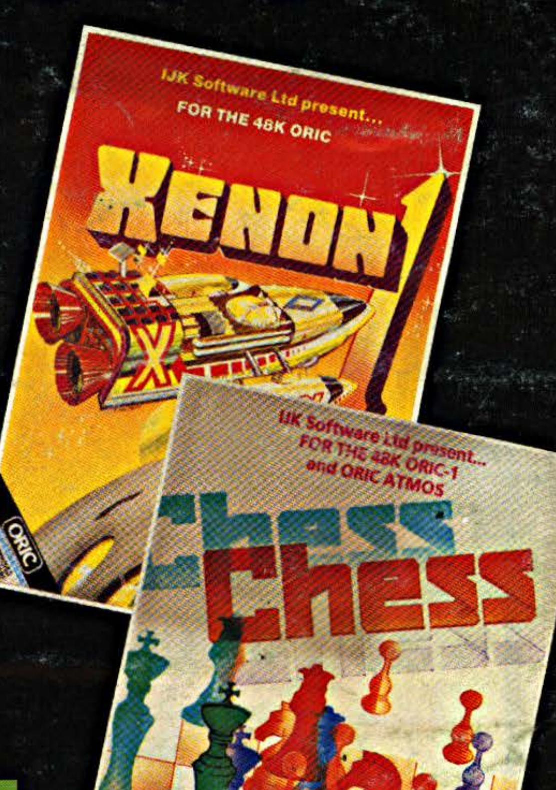
Adresser votre commande à :

A.S.N. DIFFUSION (Département composants) - B.P. 48 - 94470 BOISSY-SAINT-LÉGER

logiciels





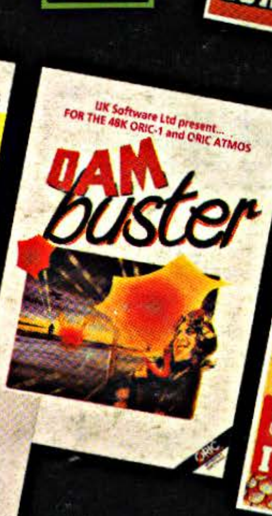


3 Nouveautés











POUR ORIC

Disponibles chez ASN